Paper

Dynamic Contracting of IP Services – System Architecture and Prototype

Piotr Arabas and Mariusz Kamola

Warsaw University of Technology, Institute of Control and Coputation Engineering, Warsaw, Poland
Research Academic Computer Network, Warsaw, Poland

Abstract—Proposition of architecture and implementation of prototype system for dynamic contracting IP services is presented. The system serves requests issued by users demanding setting up network service of specified parameters. DiffServ technology together with traffic engineering and admission control are used. Implementation details and results of tests are described. Necessary extensions and possibility of commercialization of such a system are discussed.

Keywords—DiffServ, dynamic contracts, IP network, quality of service.

1. Introduction

The growing demand for bandwidth is unquestioned. It results mainly from popularity of equipment allowing capture of high quality video, voice and still pictures as well as software assisting in editing and further distribution of such data. Potential users may be individuals using it mainly for entertainment (VoD, interactive games, etc.) but also companies and institutions applying them for teleconferences, distant learning or telemedicine. As such services seem to be more involving resources than others, it is natural to model influence of users' requests on them. This leads to proposition of a system for bandwidth reservation that can use information like requested start time and QoS requirements for a service to fulfill, if possible, user demands. The paper presents the project of a single domain system, as well as working prototype of limited functionality, and results of experiments.

In the last years many research projects dealt with the problem of providing QoS in IP networks, mostly assuming the existence of some signaling scheme, like the one presented in [1]. Some of them focused mainly on technical aspects, the most interesting being TEQUILA, MESCAL, AQUILA and EuQOS. In short all of them tried to build additional control layer allowing providing QoS guarantees to end users with the help of DiffServ and various methods of traffic engineering depending if they were to operate in a single (MESCAL, AQUILA) [2] or multiple domains [3], [4]. Common for all projects was an admission control logic, which seems to be necessary when providing QoS guarantees using limited resources, and poses important challenge if it is to be working in efficient, scalable way. Another stream of work was dedicated to economics of such systems, with M3I, QOSIPS and CoCOMBINE being probably the most representative projects [5], [6]. Abovementioned projects were EU-funded and had purely academic flavour. On the commercial side there are industrial standards like MPLS, DiffServ, various blends of RSVP and other reservation, monitoring and policy enforcement (e.g., SNMP, COPS, Diameter) related protocols. There is, however, no complete solution for bandwidth reservation, traffic engineering, network management and business processes; ITU NGN framework being too general and usually implemented in fragments. On the other hand vendors of network equipment offer, often expensive, software for management of their devices covering not only monitoring and maintenance but also network planning and optimization – Cisco IP Solution Center being one of examples [7].

2. Functionality and Architecture of the System

Network operators have not adopted any of the previously mentioned products and standards to build IP-based reservation systems as most of the research projects resulted only in general conception of the architecture or a limited prototype for demonstration in the laboratory. Commercial platforms, being more mature and suitable for real world application are costly, also because they must be tailored to customer needs, and maintained afterwards. All that brought the authors to the idea of developing at NASK a prototype of a reservation system – firstly, for research purposes, but with following commercial application on mind. The main function of the system is the ability to contract network resources of specified parameters: duration, source and destination points, QoS and price. The architecture is centralized as reservations are made in a domain possessed by single operator, however the inter-working is not precluded if other domains will use similar systems. Hierarchical organization is envisaged to provide scalability.

Main functional blocks and modules of the system are depicted in Fig. 1. The central element of the system is negotiations block containing logic for interaction with users and wholesale operators (needed when connection traverses

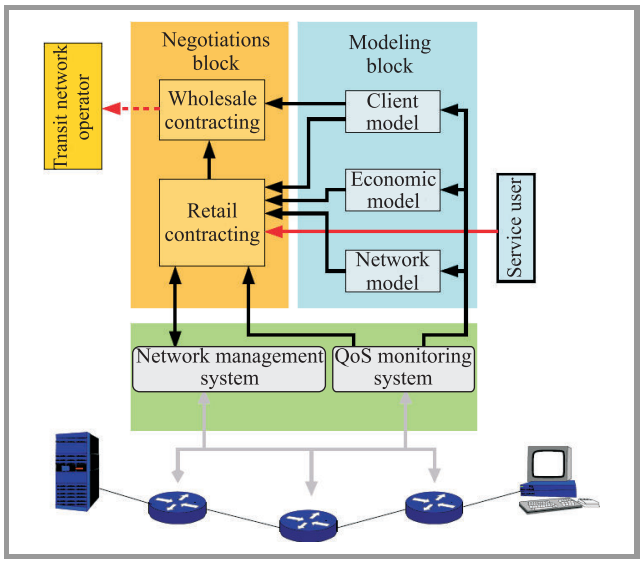

Fig. 1. Architecture of the network reservation system.

other domains). Modeling block and monitoring and actuation block serve negotiations block with necessary data and allow execution of the contract.

2.1. Negotiations Block

Negotiations block is the only block that interacts directly with users accepting or rejecting their requests. The decision whether to serve or deny a request depends on two factors:

– technical possibility of providing the service,

– profitability of the contract.

The technical possibility of providing the service is determined by the network state at the moment of the execution of the contract. As the service is usually requested in advance, technical possibilities must be predicted, taking into account:

– available network resources,

– contracts requested concurrently.

These two factors are managed by network and customer models being part of the modeling block. The first task does not pose great problem as network infrastructure is varying slowly so it is possible to tune successfully appropriate models. The other requires predicting both parameters of contracts and their utilization by users, which is much more complicated.

The definition of profitability takes into account not only simple calculation of costs and income, but also reflects long-term strategy of the provider. For example, policy of increasing margin as the network load grows will favour clients willing to pay more, and make less wealthy ones request contracts early. Acting differently, the latter ones loose chance of getting any bandwidth at a desired time. In general, policy of satisfying user requests strictly (in sense of contracts timing) is contrary to network operation economy, where costs are independent on usage, and where constant and stable utilization would fit the providers best. Combining the "maximize income" and "minimize costs" goals into a consistent and competitive pricing scheme is a demanding task.

Estimation of profitability becomes even more complicated when operator has possibility to dynamically make wholesale contracts for transit links. In such situation network resources of the operator are not constant as they may be periodically renegotiated. Such renegotiation is triggered by analysis of scheduled contracts but, reciprocally, strategy of contracting depends on transit costs. Cross dependency may be solved by assumption that wholesale contracts are valid for time longer than decision horizon of retail contracting. So, the module responsible for retail contracts may treat wholesale contracts as given and make decisions using only the economic model. Module responsible for wholesale contracts analyzes retail contracts (both accepted and rejected) ex post and, if improvement is possible, renegotiates them with wholesale bandwidth providers. This way two levels of contracting are decoupled securing global scalability of the system.

2.2. Modeling Block

Modeling block consists of three specialized models: network model, economical model and customer model. Using these models it is possible to feed negotiation algorithms with predictions necessary for computing optimal contracting strategies. Network and customer models are adjusted using traffic and contract data available in abundance from network monitoring and customer relationship management (CRM) systems. Economic model is more static due to its expert character. The task of network model is assessing network state while some set of contracts is active. A single contract C_i may be described by the following set of parameters:

– requested start and termination time – α_i and ω_i,

– requested source and destination points – o_i and t_i,

– requested average and peak bandwidth – b_i and p_i,

– requested maximum delay d_i, and its variance j_i,

– requested loss rate r_i,

– a vector of parameters $\mathbf{f}$ describing actual way of utilizing the contract by the user; it may contain parameters like the ones above, but also more specific ones w.r.t. traffic engineering: e.g., effective bandwidth [8].

Network model predicts how to configure network equipment to provide contracts in optimal way. The output of the model are variables describing working conditions of all links as well as prediction of quality parameters for contracts being available bandwidth, delay with variation, loss rate and path selected for transmitting data.

Network model refers to network topology and technology (including QoS provisioning technology) to predict utilization of resources and so state of the network. These data combined with set of contracts selected for execution constitute optimization task where the network topology, contracts and traffic multiplexing rules are the constraints. The method of providing contracts (depending on the technology of the network) defines constraints but also the performance index, e.g., when using DiffServ the contracting strategy imposes class of service but also requires to manage resources – select appropriate paths, balance loads on links etc.

The way of modeling interactions between transmitted streams depends on nature and number of these streams. One of most advanced approaches may be effective bandwidth theory [8], however it requires detailed knowledge of traffic characteristics and is applicable only in case of large number of concurrent streams. In smaller scale simplified models (mainly pessimistic) or even network simulators (e.g., ns-2 [9]) may be used. Time necessary for completing simulation precludes its on-line use, however it could be helpful for periodic tasks as traffic engineering and optimization of contracts parameters. The main reason for modeling the network is admission control – in such case a list of active contracts with paths selected for them is an additional constraint for the optimization task.

Client model predicts real user traffic pattern versus the resource consumption as declared by a user. It is fed with abundant historical traffic data. This way traffic engineering may be more effective as its input is closer to reality. It is important, however, to remember that degree of utilization of requested bandwidth is not constant. It usually depends on kind of service and it may also vary with price of service as willingness to pay may reflect meticulousness of users. This results in coupling between pricing and client modeling making optimization task more complex. The handling of such phenomenon is to use pessimistic approach for accepting contracts and tune the approximation when statistics become available.

Economic model provides relation between quantities describing services, costs and incomes generated. Association between costs and services is not straight as some costs are generated directly while others inflicted by sub-services used.[1] Fair distribution of costs should prevent unjustified lowering price of some services (while other subvent them) and so protect competition. In practice, however prices usually do not result from costs, but models of costs distribution are used internally in the enterprise to assess profitability and, when optimizing pricing strategy, become part of performance index or constraints.

As stated before, services may be provided with use of resources bought in wholesale contracts from higher tier providers. The fluctuation of wholesale prices may influence costs of services, however it is assumed that their time scale is much longer than horizon of operational decision making which thus may be performed for static set of wholesale contracts.

Economic model describes also dependency between demand for services and their prices, which may be done with well known parametric models like Cobb-Douglas model that uses elasticity to connect changes in price with varying demand or utilization [11]. Thanks to abundance of data available for dynamically contracted service model parameters may be regularly updated, preferably for models prepared separately for various market segments. Initial values of parameters must be, however, chosen when data are scarce: then expert models are to be used.

The output of economic model is used in negotiations block supporting process of wholesale and retail contracting by determining performance index. It must be stated that performance index may reflect various short term targets like maximizing volume of contracts, minimizing number of rejected requests or contracts violating QoS guarantees (to maximize number of satisfied users), but always in longer horizon it supports the same economical targets.

2.3. Monitoring and Actuating Block

Monitoring and actuating block constitutes adaptation layer between modeling and negotiations blocks and network equipment. The QoS technology is fundamental for contract providing, it must however be implemented in network equipment, and to manage equipment of various vendors uniformly, the set of communication procedures is necessary. For the same reason this layer consists of specialised procedures for monitoring state of equipment and QoS level of contracts.

Network management subsystem provides means for setting contracts, which, for DiffServ and MPLS technology, requires setting of:

- flow classification,
- traffic profiling and shaping,
- prioritizing traffic via class queues,
- MPLS queues.

Communication with network equipment may be implemented with standardized protocols like SNMP and LDAP or using telnet and CLI. The efficiency and complexity of both approaches is similar, the main problem being laborious checking of correctness of configuration.

[1] Establishing a fair cost-splitting scheme for common networking infrastructure is a task of its own. A useful cost model has to care for theoretical properties (various definitions of fairness [10]) while taking into account organizational and technology constraints (e.g., granularity of monitoring energy consumption). In NASK, a proprietary cost-splitting scheme has been applied in day-to-day operation. It may serve as good starting point for further customization and development, also as regards the reservation system presented in this paper. In the scheme, lower-level common costs are allocated to higher-level services hierarchically, mostly corresponding the ISO/OSI network layer model. For practical reasons, there are numerous simplifications in the scheme, eg. the cost of maintaining a leased line is averaged for all urban, suburban and rural infrastructures, respectively, resulting in three basic prices of the kilobit-segment accounting unit. Selected details of the model are available on request from the authors of this paper.

Monitoring QoS level of services may be done in various ways, thus generating different load in the network equipment. Limited monitoring functions of network equipment can be enhanced significantly by installing additional software modules which however may adversely influence effectiveness and stability. Another solution is placing additional probes transparently monitoring traffic, however they are costly and may still introduce some additional load to the network. The compromise solution is using as much monitoring functions of network equipment as possible without lowering performance of routers and installing some probes at carefully chosen nodes (e.g., near core routers), which, together with estimation techniques should provide precise picture of the state of the network.

3. System Prototype

A prototype implementing limited set of the proposed system functionality was implemented and tested in NASK laboratory. The development started with preparation of network testbed. Next, monitoring and actuating block were implemented and, finally, application logic elements were added.

3.1. Network Testbed and QoS Mechanisms

The testbed consists of seven routers connected in a way simulating double star topology typical to providers' network. Two routers constitute the redundant core infrastructure, with two edge routers connected from both sides. Other three routers serve as client appliances — connecting two of them to the same edge router creates a potential bottleneck spot. The scheme of the network is depicted in Fig. 2. Cisco 2509 were used for core routers while edge and client routers were Cisco 1720. Using uniform equipment made it easier to implement control and monitoring functions especially after having updated operating system to the same version implementing basic set of DiffServ functionalities (traffic classification, class queues, shaping and policing). Connections between routers are provided via serial interfaces, which allows adjustment of the link speed in broad range up to 2 Mbit/s. Additionally, there are six PCs in the network, three of them hosting traffic generators and simulating users stations, while the other three acting as traffic probes, interconnecting the workstations PCs with the client routers. Control over those three computers is provided via a separate network, not shown in Fig. 2.

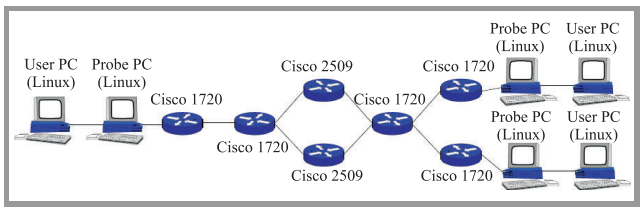

Fig. 2. Testbed topology.

Three service classes were set up in the network: nRT class designed for carrying elastic, mainly TCP traffic, RT class for traffic with strict time requirements, needing low delay and low delay variation, and best-effort class with no quality guarantee. Following DiffServ specification nRT may be associated with one of AF classes and RT is the EF class. These classes were implemented in routers by means of Cisco CBWFQ (nRT and best-effort class) and priority (RT) queues, which allowed minimizing delay thanks to absolute priority over other classes. Client routers were responsible for classifying incoming traffic to classes reflecting contracts, marking and enforcing the guaranteed bitrates by shaping (nRT) and policing (RT). Mean rates of shapers were equal to the rates requested in contracts and buffers were sized proportionally to it (amounting for approximately 1s burst). Marking and policing in client routers provides typical DiffServ scalability as core and edge routers can forward packets analyzing only their DSCP.

Apart from all analogies to the production network it is important to remember about limitations of the testbed resulting from simplified topology and use of basic equipment but also very limited length of links. Specifically, propagation delays in testbed are much smaller, while queuing delays may be, especially during heavy congestion, longer than real. To use such installation efficiently it is necessary to scale experiments, e.g., it is not possible to transmit several HD video streams as link capacity of 2 Mbit/s is several times too small; however after scaling them down to low quality it is possible to send a few streams simultaneously. It is also possible to transmit one stream of better quality and compare results of changing the scale.

3.2. Elements of Monitoring and Actuating Block

To allow monitoring and management of network equipment, a library of communication procedures constituting lower abstraction layer of monitoring and actuating block was developed. Basing on the experience of former projects (e.g., AQUILA) communication via telnet was chosen which, together with using Perl for parsing commands, made library relatively efficient and robust. The procedures are designed primarily for configuring DiffServ-related functions in Cisco routers, however possibility of further adaptation was taken into account by dividing them into two groups. Low level procedures cover basic configuration functions of Cisco routers and are used to provide functionality needed by negotiations block for setting up and managing contracts. This way porting is possible by changing low level procedures only.

Efficiency of implementation, so important when building an on-line system, is however difficult to attain together with reliability as typically network equipment has not been designed with on-the-fly reconfiguration on mind. This makes communication not only time consuming but, worse, it makes verification of configuration difficult – often it is necessary to analyze router output to know if the resulting configuration is consistent. Repeating such pro-

cedure frequently (e.g., after every command sent) introduces unacceptable communication and computing overheads. The solution is performing only basic, mainly syntax, checks frequently and confronting routers configuration periodically with information stored in system database. Basic procedures supporting such verification are part of the library prepared. The functionality of the library supports:

- reading complete configuration of routers,
- sending CLI commands,
- setting clock rates of serial links,
- reading state of interfaces,
- managing access lists,
- managing policers and shapers,
- managing DiffServ AF queues,
- managing route maps,
- managing contracts (i.e., configuring classifiers, queues, shapers or policers etc.),
- reading statistics of queues, shapers, policers, interfaces,
- reading statistics of bandwidth allocated to specific contracts.

3.3. Negotiation and Contracting Logic

Implemented negotiation and contracting algorithms are much simpler than described in Section 2, in fact, the prototype presented in this paper is a proof of concept for the complete reservation system. A rich set of batch programs for managing contracts has been implemented, which use text files to keep track of contracts, and a communication library described in Subsection 3.2 to configure and monitor routers. Specifically, text files are used as the interface between programs, passing information about

- active contracts,
- traffic statistics,
- network topology and traffic engineering paths.

Data access classes are designed so that they can be reused in a future production release of the system, built upon a relational database. Simplifications are acceptable because presented implementation is used only for testing in the laboratory environment; furthermore, implemented programs include some functions which are typical for testing, e.g., program serving contracts not only provides CAC and configuration functions but also, after having successfully set the contract up, it starts traffic generator, thus simulating user activity. In a case when requested contract cannot be served, the system computes estimated time when it is likely that resources will be available and, again, for simplification of experiments reservation program, waits and reissues the request automatically. Another simplification consists in removing expired contracts periodically by searching active contracts. In a real system such solution would lead to extensive computational overhead, and probably accounting errors; it is however acceptable in testbed experiments with limited number of contracts.

CAC functions implemented in negotiation logic use direct feedback from the system to assess the possibility of accepting a contract. First, an appropriate path is selected from statically configured set, then available bandwidth on all links building the path is checked, to find out a bottleneck in a manner similar to RSVP operation [12]. Such a procedure influences scalability obviously, however it is reasonable in the case of centralized control as it allows better utilization of bandwidth than local rules allocating usually preallocated amount of bandwidth [13]. Various measurement-based admission control algorithms use advanced models of aggregate flow, e.g., effective bandwidth [14] and require much attention to extracting precise parameters of both measured aggregate and new flow. Aggregate flow modeling involves collecting difficult to obtain data (e.g., short term averages or peak rates) and complex computations, while estimating parameters of a new flow assumes knowledge of user behavior [15]. The prototype nature of presented system justifies simplifications, specifically, the new contract is admitted after positive evaluation of the following formula:

$$c_{nRT} - \sum_i b_i^* - \gamma \sum_i b_i - \alpha b_{i+1} > 0 \ , \qquad (1)$$

where: c_{nRT} – bandwidth reserved for nRT class, $\sum_i b_i^*$ – bandwidth occupied by all active contracts (read from router statistics), $\sum_i b_i$ – bandwidth requested by all active contracts, b_{i+1} – bandwidth requested by the contract being subject of CAC procedure. Mixing coefficient α allows overbooking (when lower than 1.0), and should be experimentally estimated, while coefficient γ provides some margin over smoothed statistics to be utilized in bursts. In this way it accounts for burstiness and peak rate in a manner similar to used in typical formulations (e.g., [16]) as in experimental set-up buffer size is proportional to average rate declared in the contract. The role of coefficient α is to envisage some level of overbooking at the moment of processing new request, and utilize available capacity more aggressively, which is important in case of limited number of flows. If the formula (1) result is negative the request is blocked and time to wait is computed taking into account parameters of other contracts.

4. Efficiency Tests

Extensive tests were performed to verify various aspects of system architecture and technology used. The following have been checked: efficacy of traffic prioritization,

Table 1
Duration of equipment reconfiguration process

Subject of the test		Number of contracts	Time of setting up [s]		Total time [s]
			first contract	last contract	
Communication library	version 1	100	4	38	2221
	version 2	100	3	6	352
Management program		100	6	9	852

effectiveness of contract management functions, stability of equipment during frequent reconfigurations and CAC algorithms operation.

4.1. Efficiency of Contract Management

The aim of tests was assessment of contracting functions efficiency. The experiments were carried out in two stages: first, the communication library was tested followed by tests of the whole contract management program. In both cases no data were transferred through the network – only requests were processed and routers reconfigured, which allowed measurement of time required to set up the contract by software and equipment. Two versions of contracts setup procedure were tested: the first one adjusts parameters of all already present priority queues while processing every new request, to ensure proportional distribution of bandwidth in nRT class; the second one allocates constant fraction of requested bandwidth, which also results in proportional share and is less time consuming (only one queue is configured each time).
Results of tests are presented in Table 1. Some fluctuations of time were observed, they are probably caused by traffic interference and varying load on the management server. The first version of algorithm is much slower as repeated reconfiguration has computational complexity of $O(n^2)$ for n contracts. Operation of the algorithm in second version is acceptable, it must be however pointed out that periodical recomputing of contracts and reconfiguration of routers may be necessary in case when number of contracts grows over value assumed.
Measurement of execution time for management program was carried out in the similar manner, using second (faster) version of contracting procedures. Setup time overhead introduced by the program is 2 to 5 s and grows with a number of contracts, which is the effect of necessity of analyzing all contracts.

Conclusions. Reconfiguration time is acceptable, however in case of short-time contracts (e.g., lasting several minutes) may be considered annoying by some users. The main limitation lies in routers that process requests sequentially and with limited performance (typically 10 to 15 requests per minute). This time may be shortened by optimization of programs and, first of all, by replacing routers used in testbed with equipment more advanced and better suited for the role (e.g., specialized traffic shapers, etc.). It is also hoped that most reconfiguration actions take place at network edges, being the result of managing requests in a distributed way, and therefore no high request processing volume is required for such a single edge device.

4.2. Possibility of Providing Adequate QoS for nRT Class

Test described in this subsection were devoted to checking ability of providing connections with adequate QoS, namely:

- QoS guarantees in nRT class,
- possibility of bandwidth sharing by nRT and best-effort classes,
- estimating maximum number of contracts which can be served by single router.

It is expected that flows will be transmitted separately with bitrate similar to shaper bandwidth and smoothed by shapers. In periods of limited activity in nRT class best-effort class may occupy additional bandwidth, it should not however influence higher class performance. Estimation of maximum number of contracts supported by routers is crucial to further commercialization of the system, but limitations of equipment used in testbed must be taken into account.
Four experiments were performed, differing in number and type of flows:

- 14 flows, each requesting 128 kbit/s in nRT class,
- 15 flows, each requesting 128 kbit/s in nRT class,
- 14 flows, each requesting 128 kbit/s in nRT class, plus elastic best-effort traffic
- 30 flows, each requesting 64 kbit/s in nRT class.

Contracts were set up prior to traffic generation to prevent configuration disturbing flows. Link rates were configured on 2 Mbit/s level. Four variants of experiments were prepared. In the first variant bandwidth requested by nRT flows is 1792 kbit/s, or 87.5% of total capacity, in the second – 93%. The third variant was designed to test possibility of filling up free bandwidth with best-effort traffic – nRT flows occupied 87.5% of link capacity. The last experiment was devoted to checking maximum number of flows that can be transmitted concurrently.

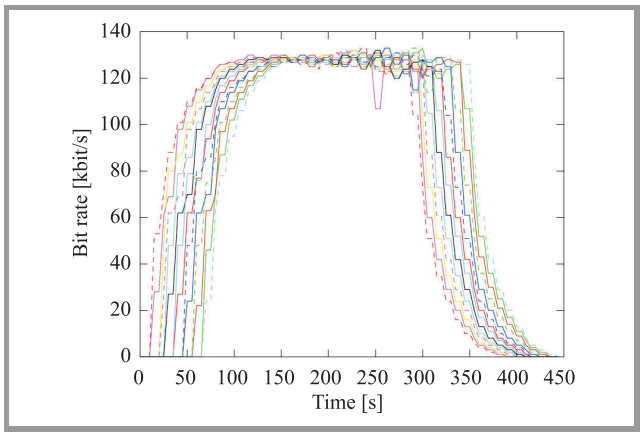

Fig. 3. Average bit rate of 14 nRT flows.

Results of experiments are presented in plots depicting mean bit rates of flows during experiments (30 s intervals were used). Figure 3 reflects results of the first experiment – it may be observed that bandwidth is distributed consistently among flows and guarantees are kept (maximum fluctuations are no higher than 2% which seems to be acceptable for TCP traffic). The second experiment (see Fig. 4) was not so successful – it seems that more than 7% of headroom is necessary to accommodate fluctuations of TCP traffic. The third experiment (see Fig. 5) was

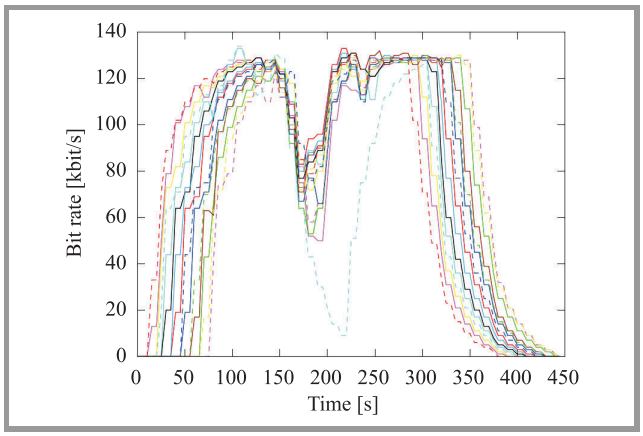

Fig. 4. Average bit rate of 15 nRT flows.

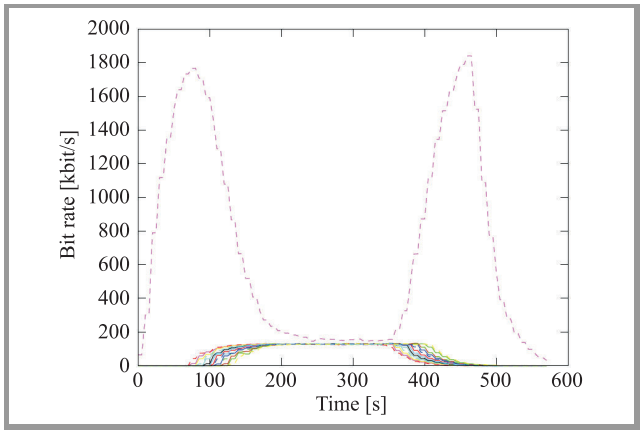

Fig. 5. Average bit rate of 14 nRT flows and best-effort flow.

designed to verify preemptive operation of class queues. It may be observed that best-effort class utilizes link fully when no higher class traffic is generated, however it limits its throughput immediately when first nRT flow has started. Additionally it is worth to notice that existence of best-effort traffic allows higher network utilization than in case of transmitting only QoS guaranteed traffic (1880 kbit/s versus 1792 kbit/s in the first experiment).

The outcome of the last, fourth experiment was negative – transmission was broken due to malfunction of router during setting up approximately twentieth contract. This way it may be estimated that maximum number of class queues and so contracts supported by this lower class Cisco router is some 20.

Conclusions. DiffServ implementation provided in Cisco IOS allows effective prioritization of a limited number of flows. Existence of best-effort traffic not only does not degrade higher classes but also allows better utilization of links. Typical use scenarios and resulting numbers of contracts demanded must be analyzed carefully to define target users of dynamic contracts and select equipment (and so, costs) apt for such an enterprise.

4.3. Analysis of Transient States During Setting up of Contracts

Tests carried out in these group were designed to check possibility of sharing the link by RT and best-effort traffic. Two issues were considered:

– prioritization of RT traffic,

– transient network states caused by configuring new contract and their influence on active contracts.

Possibility of preempting best-effort traffic by UDP RT traffic was tested in experiments. Use of UDP traffic lacking congestion control mechanism allowed credible measurement of packet loss. Checking how QoS parameters of existing contracts change during reconfiguration of routers is important because only when there is virtually no influence, dynamic contracting is possible. Otherwise, when reconfiguration disturbs existing transmission, system applicability will be highly limited.

Table 2
Packet loss rate and delay statistics for UDP priority transmission

		Priority class	
		1 flow	2 flows
Best-effort class	4 flows	**0** *20/39/85*	**1.3** and **7.8** *20/44/86*
	6 flows	**0** *20/42/85*	**6** and **0.5** *20/45/86*

To test prioritizing in RT class, a number of video streams in H.263 format with bit rate 256 kbit/s were transmitted simultaneously. Picture resolution was 176×144 pixels –

the generic setting for mobile phone application. H.263 is CBR codec, however peaks up to 1.4 Mbit/s were observed. During experiments one or two video streams were transmitted in RT class, and four or six in best-effort class. Bandwidth allocated to RT queue was 90% of total link bandwidth. Table 2 presents video streams parameters: percentage of packet loss in subsequent transmissions in bold face, and total delay statistics for priority traffic in italics (minimum/mean/maximum delay in ms).

Conclusions. Tests confirmed that priority traffic is transmitted correctly. In case of a single flow no packets are lost and delay is kept at acceptable level. Some fluctuations of delay show that packets are successfully buffered which prevents dropping. When two flows are sent simultaneously some loss may be observed which is result of using relatively short queue (8 packets). Short queue however limits maximum delay to the level similar as in case of one stream. Very limited influence of best-effort traffic must be noted which makes the tested routers a good choice when real time traffic prioritizing is needed.

The following experiment aimed investigating transient states which can occur during reconfiguring routers to set up new contract. It was suspected that additional load to the router caused by its reconfiguration may influence existing contracts. To verify this, packet loss and delay during reconfiguration were carefully observed. Test traffic consisted of 6 VoIP streams transmitted in best-effort class and 3 identical streams in RT class. All streams use G.723.1 protocol sending frames of 24 B every 30 ms. To test the impact of router reconfiguration, an additional fourth contract is set up after 200 s from starting the first nine ones. Setting up contract comprises all activities from registering it in contract base to configuring access list in the routers. After completing these tasks transmission is started in 210th second from beginning of the experiment.

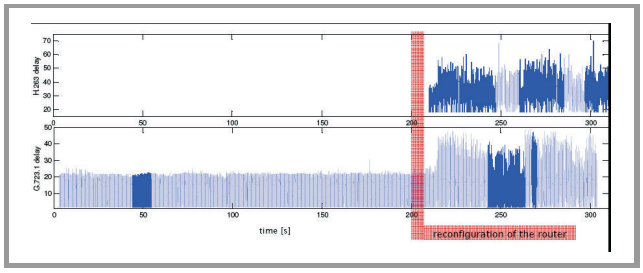

Fig. 6. Delay of original 3 VoIP transmissions (lower graph) and additional VoIP transmission (upper graph).

Figure 6 presents results of the experiment. Lower graph depicts delay of first three VoIP streams sent in RT class while in the upper graph delay of additional fourth transmission may be observed. Thanks to the delay between reconfiguration ($t = 200$ s) and start of transmission ($t = 210$ s) it may be easily noticed that reconfiguration influences delay in negligible degree. After starting fourth stream delay grows significantly due to higher load, however it still remains on appropriate level (well below 100 ms).

Conclusions. Reconfiguration of the router does not influence data transmission, so dynamic contracting is possible provided the router is not overloaded. In case of overload configuration functions (*management plane*) are influenced first than data forwarding (*data plane*).

4.4. Admission Control Algorithm

The subject of the test was checking strategy of contract admission described in Subsection 3.3. and tuning its parameters. The aim of this algorithm is maximizing network utilization while keeping QoS guarantees in nRT class. As flows transmitted in nRT class use TCP protocol, main aim is providing mean bit rate close to requested value. The typical use of such a service may be WWW browsing, which in turn may be modeled by *on-off* generator with exponential *off* periods and Pareto sizes of data to transmit. Two sets of generator parameters were used during the experiment:

1. $k = 1.2$, $o = 512$ kbit/s, $\lambda = 2$ – mean bit rate approx. 70 kbit/s,
2. $k = 1.2$, $o = 512$ kbit/s, $\lambda = 5$ – mean bit rate approx. 200 kbit/s.

The k parameter is shape coefficient of Pareto distribution[2], o is maximum bit rate of generated flow and λ is inverse of mean *off* time. In both experiments 30 contracts lasting 300 s are requested every 5 s. Each of them requests 512 kbit/s which is its peak rate. Total declared bandwidth (15 Mbit/s) exceeds available bandwidth set to 2 Mbit/s. Actual total mean bandwidth of all contracts is much lower. It is 2.1 Mbit/s in the first and 6 Mbit/s in the second case, however is still beyond available bandwidth limit, so for correct functioning of the network efficient admission control mechanism is necessary.

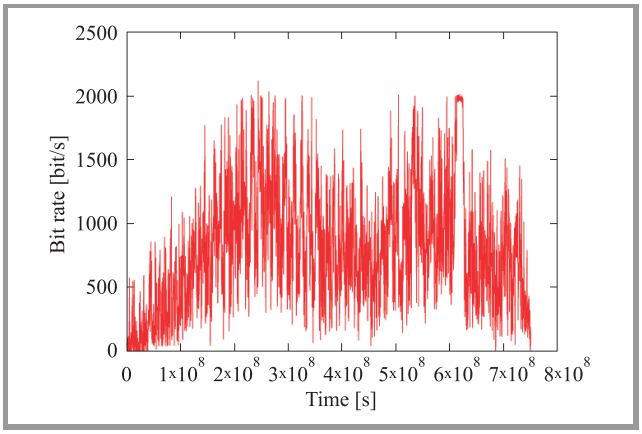

Fig. 7. Short term average bit rate for traffic generated with first set of parameters.

Figures 7 and 8 present average bit rate graphs of traffic captured on the interface of receiving machine during

[2]Pareto distribution PDF: $f(x) = kx_m^k/x^{k+1}$. Parameter x_m is in both cases equal 1200 B which seems to be reasonable estimate of minimum amount of data sent in single transmission.

the first variant of experiment. Averaging period was 0.3 s and 30 s respectively. Both graphs show the decrease of bit rate in the middle of the experiment, which is the result of postponing of nearly half of contracts for 300 s, so they could start after completion of the first group. Short term averages show huge burstiness of traffic, which for a short while reaches the declared maximum. Analysis of graph with longer averaging period reveals that network is not fully utilized (in fact 15 active contracts need $15 \cdot 70$ kbit/s = 1.05 Mbit/s) and parameters of admission mechanism seem to be too conservative. It must be noted however that in terms of declared bandwidth network offers high degree of overbooking – 15 active contracts means that $15 \cdot 512$ kbit/s = 7.68 Mbit/s were simultaneously contracted, while 2 Mbit/s were available.

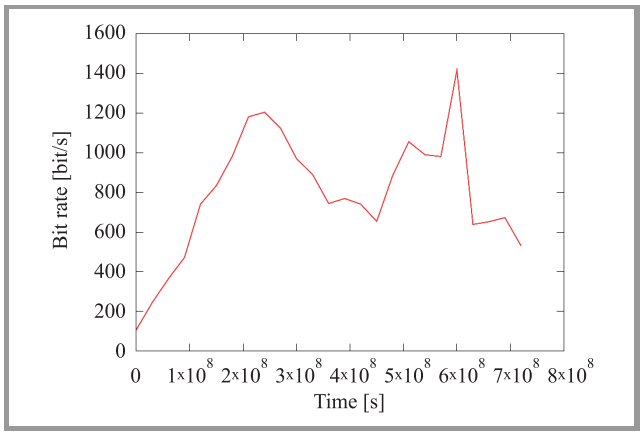

Fig. 8. Long term average bit rate for traffic generated with first set of parameters.

Bit rates observed while executing the second version of the experiment are depicted in Fig. 9 and 10. Again, short (0.3 s) and long (30 s) term average bit rates are presented. Similarly to the first variant, nearly half of contracts were postponed, however generated bit rates were much higher. By comparing long-term averages it is easily noticeable that number of accepted contracts is much too high, which results in congestion. In case of such overload by bursty traffic admission strategy is too liberal, which results not only in limiting bandwidth of flows, but also in its unfair distribution among contracts (flow bit rates span from 30 kbit/s to 150 kbit/s).

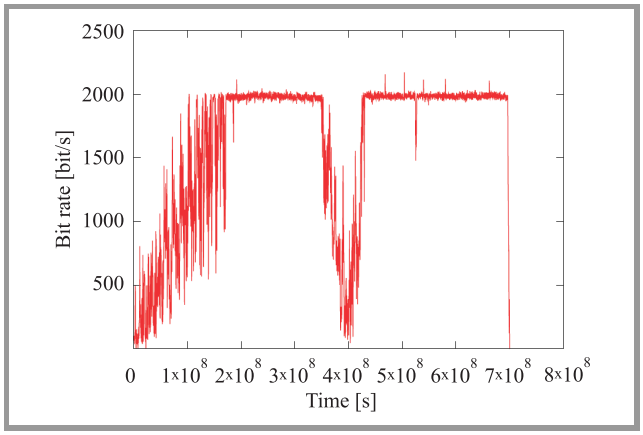

Fig. 9. Short term average bit rate for traffic generated with second set of parameters.

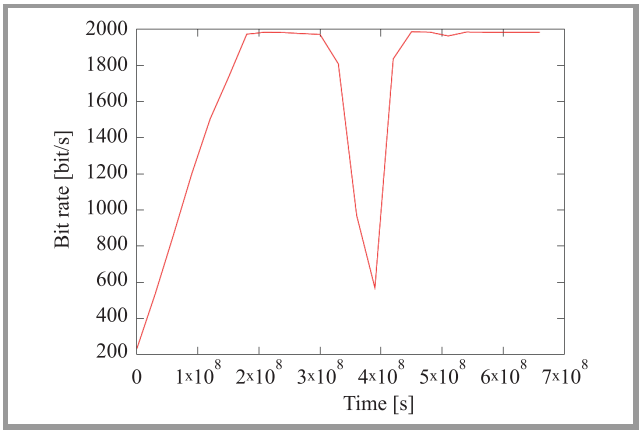

Fig. 10. Long term average bit rate for traffic generated with second set of parameters.

Conclusions. Efficient admission control using model (1) with constant coefficients for various kinds of traffic is impossible. Additional problem is using 30 s averages to monitor network state which introduces too much delay in control loop (in analyzed case new request arrived every 5 s).

5. Summary

Experiments performed were designed to check main prerequisites of dynamic QoS contract management system, namely:

- efficiency of DiffServ flow prioritizing under heavy load of offered traffic and configuration commands,
- ability of network equipment to frequent reconfiguring of contracts,
- applicability of admission algorithm proposed.

In authors' opinion results of tests are optimistic and justify continuation of the work: even relatively simple equipment allows to efficiently configure and prioritize contracts. Data and management planes are appropriately separated and do not influence each other. The number of contracts supported is limited, however it is reasonable for serving a household.

Tests also revealed weak points, those being slow and unreliable communication with routers, limitation of simultaneously active queues and difficulty of CAC algorithm tuning. The efficiency of configuring the equipment may be improved by implementing some functions in parallel and providing better algorithms for checking configuration and monitoring network parameters (switching to communication via SNMP may be helpful). To build heavily loaded system equipment of greater efficiency (and possibly different technology) is however necessary. On the other hand

improvement of admission strategy is possible mainly by modification of algorithms by:

- estimating real bit rates between measurements (it may be done by including input from contract database),
- introducing additional parameters determining traffic characteristic into contract request (e.g., some measures of burstiness) to tune algorithm parameters individually,
- providing more QoS classes to make traffic transmitted in particular class more consistent.

Summing up, the existing technology allows to implement the system providing dynamic contracts with QoS guarantees. Neither existing networks structure nor scalability and amount of work necessary to develop such a system constitutes major obstacle in its implementation and introduction. Furthermore, such a system will naturally assist operators in offering new services to users, it could also be attractive to users offering them quality guarantees contracted dynamically, so (probably) cheaper than today. The real problem is no real need for such services observed on the market.

Acknowledgement

The paper was partially supported by Polish Ministry of Science and Higher Education grants N N514 416934 and PBZ-MNiSW-02/II/2007.

References

[1] X. Wang and H. Schulzrinne, "RNAP: a framework for congestion-based pricing and charging for adaptive multimedia applications", *IEEE J. Sel. Areas Commun.*, vol. 18, no. 12, pp. 2514–2529, 2000.

[2] M. Howarth (Ed.), "Initial Specification of Protocols and Algorithms for Inter-domain SLS Management and Traffic Engineering for QoS-based IP Service Delivery and their Test Requirements", Deliverable D1.2,MESCAL, 2004.

[3] D. Goderis (Ed.), "Functional Architecture Definition and Top Level Design", Deliverable D1.1, TEQUILA, 2000.

[4] J. Enríquez and J. Andrés (Eds.), "Definition of Business, Communication and QoS models – Intermediate", Deliverable D.1.1, EuQOS, 2005.

[5] P. Arabas, M. Kamola, K. Malinowski, and M. Małowidzki, "Pricing for IP networks and services", *Inform., Knowl., Sys. Manag.*, vol. 2, pp. 153–171, 2003.

[6] I. Constantiou (Ed.), "ISP Business Model Report", Deliverable 7.1, M3I, 2002.

[7] "IP Solution Center – MPLS VPN", white paper, Cisco Inc., 2003.

[8] F. P. Kelly, "Notes on effective bandwidths", in *Stochastic Networks: Theory and Applications*, F. P. Kelly, S. Zachary, I. B. Ziedins, Eds. Oxford University Press, 1996, pp. 141–168.

[9] NS2 [Online]. Available: http://www.isi.edu/nsnam/ns

[10] C. Courcoubetis and R. Weber, *Pricing Communication Networks: Economics, Technology and Modelling.* Chichester: Wiley, 2003.

[11] G. L. Lilien, P. Kotler, K. S. Moorthy, *Marketing Models*. Prentice Hall, 1992.

[12] R. Braden (Ed.), "Resource ReSerVation Protocol (RSVP) – Version 1 Functional Specification", RFC 2205, 1997.

[13] A. Bak, W. Burakowski, F. Ricciato, S. Salsano, and H. Tarasiuk, "Traffic handling in AQUILA QoS IP network", in *Quality of Future Internet services* of *Lecture Notes in Computer Science 2156*, Springer 2001.

[14] J. Ko, "A decision theoretic approach to measurement-based admission control", in *Proc. IEEE Int. Conf. Commun. ICC-2007*, Glasgow, Scotland, 2007, pp. 742–747.

[15] K. Kanonakis, H. Leligou, and T. Orphanoudakis, "Online flow characterisation for measurement-based admission control: a practical perspective", *Int. J. Res. Rev. Comput. Sci.*, vol. 1, no. 4, pp. 149–157, 2010.

[16] S. Floyd, "Comments on Measurement-based Admission Control for Controlled-Load Services", Tech. Rep., 1996.

Piotr Arabas received his Ph.D. in computer science from the Warsaw University of Technology, Poland, in 2004. Currently he is assistant professor at Institute of Control and Computation Engineering at the Warsaw University of Technology. Since 2002 with Research and Academic Computer Network (NASK). His research area focuses on modeling computer networks, predictive control and hierarchical systems.

e-mail: parabas@ia.pw.edu.pl
Institute of Control and Computation Engineering
Warsaw University of Technology
Nowowiejska st 15/19
00-665 Warsaw, Poland

e-mail: Piotr.Arabas@nask.pl
Research Academic Computer Network (NASK)
Wąwozowa st 18
02-796 Warsaw, Poland

Mariusz Kamola received his Ph.D. in computer science from the Warsaw University of Technology, Poland, in 2004. Currently he is assistant professor at Institute of Control and Computation Engineering at the Warsaw University of Technology. Since 2002 with Research and Academic Computer Network (NASK). His research area focuses on economics of computer networks and large scale systems.

e-mail: mkamola@ia.pw.edu.pl
Institute of Control and Computation Engineering
Warsaw University of Technology
Nowowiejska st 15/19
00-665 Warsaw, Poland

e-mail: Mariusz.Kamola@nask.pl
Research Academic Computer Network (NASK)
Wąwozowa st 18
02-796 Warsaw, Poland