

Evaluation of polling systems influence on quality of service in wireless computer networks

Igor V. Kalinin, Liudmila A. Muravieva-Vitkovskaya

CS Department
ITMO University
St. Petersburg, Russia
kalinin@cs.ifmo.ru

Abstract— Polling systems in wireless computer networks review is provided. Classifications are offered for queuing system types, queuing disciplines, and application service disciplines. The results of a comparative analysis of various polling systems on mathematical models and recommendations for their application are presented.

Keywords— queueing systems, simulation, computer network

I. INTRODUCTION

The quality of the operation of broadband wireless computer networks is largely determined by the mechanism of an ordered centralized poll of terminal stations, called polling. The models of polling systems adequately describe the operation of broadband wireless networks under the control of Wi-Fi and Wi-MAX protocols with a centralized coordination function (PCF). Despite the complexity and high cost, the equipment supporting centralized management most effectively uses the most valuable resources of a broadband wireless network: frequency resource and bandwidth. The PCF mechanism allows to plan the order of access of stations to the data transmission medium, to control the operation of the radio cell, changing its parameters depending on the situation. From the specific polling system underlying the PCF mechanism, the efficiency of the wireless broadband computer network depends largely. Since the IEEE 802.11 standards do not describe a specific method for polling terminal stations, the way to create a poll list and the policy of queue shots for sending, this article focuses on the development and analysis of various polling systems.

II. MAIN SECTION

In this paper, we perform a comparative analysis of the effectiveness of various ways to manage the flow of applications using the models, methods and tools developed by the authors. As the basic model of polling, a single-channel queuing system with a heterogeneous flow of applications is used, the processes of receipt and maintenance of which are described by exponential laws and laws of an arbitrary kind, respectively.

To conduct a comparative analysis, a lot of buffering and service strategies are conveniently classified according to the following characteristics:

1. the existence of priorities between applications of different classes;
2. the mode of supplanting applications from the buffer and assigning requests for services;
3. the rule of supplanting applications from the buffer or selecting service requests;
4. the possibility of changing priorities during the period of the system operation.

Depending on the presence or absence of priorities between applications of different classes, all buffering and service strategies can be divided into two groups, respectively: priority and priority.

By the method of crowding out the requests from the buffer (drive), we can distinguish the following classes of buffering strategies:

- without supplanting applications, which means the loss of the application that entered the system and the buffer that filled the buffer;
- with the expulsion of applications of this class;
- with the replacement of the lowest priority class;
- with the replacement of an application belonging to a group of low priority classes.

The first two classes are classified as non-priority disciplines, the other two are for priority.

In buffering discipline, the following rules can be used to preempt requests from the buffer:

- replacement random (FAR);
- expulsion of the last application, i.e. received in the system later than all;

- displacement of the "long" application, i.e. Which is in the buffer the longest.

Although in real systems the memory capacity allocated for storing data sets entering the system in the form of applications is limited, in the SRV models, the capacity of the application store is often assumed to be unlimited. This is justified in those cases when the probability of data area overflow is less than 0.001 [1], which fully corresponds to the real requirements that do not allow the loss of bids. Obviously, in this case, the characteristics of service requests are not dependent on the DB. Therefore, in the future, the focus is on service discipline.

Let us consider in more detail the classification of service disciplines.

By the method of assigning requests for service, they are differentiated to single mode; Group mode; Combined mode.

In a single mode, only one application is assigned for maintenance, i.e. scanning of drives (queues) is performed after the completion of the maintenance of each application. In the case of a group mode, a group of applications of one drive is assigned to the service, which includes in the limiting case all the applications that are in this drive. The applications of this group are successively selected from the storage drive for servicing, after the completion of servicing of all applications of this group, the next group of applications in accordance with the discipline are assigned to the service. When combined, part of the application classes is processed in a single mode, and the other part is processed in a group mode.

In the SRV, priority dispatching is implemented in the form of the following disciplines: first-in-first-out service; Service in reverse order; Service in a cyclic order.

The first two service disciplines refer to the disciplines of a single regime, and the latter to a group regime.

In the future, when considering a priority dispatch, unless otherwise specified, we will always bear in mind the discipline of the OPP, which is designated service discipline is of a non-priority nature.

Priority dispatch methods can be implemented in the following disciplines: with relative priorities; with absolute priorities ; with mixed priorities; With alternating priorities; Scheduled service.

The first three disciplines refer to the disciplines of a single regime. The service discipline with mixed priorities is a combination of relative priorities, absolute priorities and priority-free service in the order of arrival [2].

To describe the strategy, a priority matrix (PM).

$$Q = [q_{ij}] \quad (i, j = 1, H) \quad (1)$$

is used, where H is the number of classes of applications entering the system. The element q_{ij} of the matrix sets the priority of applications of class i in relation to applications of

class j and can take the following values: 0 - no priority; 1 - relative priority; 2 - absolute priority. With the help of the PM, it is possible to describe the set of disciplines, including traditional disciplines with one class of priorities.

There are various modifications of the two indicated disciplines associated with the size of the group of applications assigned to the service for one call to the queue. For example, for one access to the service queue, no more than three applications are assigned, and then a transition to another queue is made, despite the fact that the first queue has not yet served requests. The size of the group of applications assigned for maintenance can be different for different queues. In the extreme case, all orders in the queue can be assigned to the service, i. e. the processor sequentially processes the requests of the queue until it becomes empty, and only then goes to another queue.

Below we will assume that all requests in the queue are assigned for processing.

To set up an emergency, it is sufficient to indicate the sequence of numbers of application classes located in order of decreasing priority.

Prior to the PM, it is a plan (schedule) that determines the procedure for accessing the queues of applications of different classes, given in the form of a cycle ($i1, i2, \dots iG$), where ig ($g = 1, G$) is the number of the queue (class) of applications to which the inversion must be executed, and G is the length of the cycle.

Changing the number of calls to that other queue in the schedule, you can change the priority of the applications of the corresponding class.

Among the listed above, the most interesting are the disciplines with mixed priorities and scheduled services, as the disciplines of the most general kind.

In priority dispatch models, priorities assigned to applications of different classes can be maintained throughout the entire period of the system's operation or vary with time depending on the state of the system or any internal or external factors. In the first case, the priorities are called static or fixed, and in the second - dynamic or changing. Accordingly, dispatching is differentiated between static and dynamic [2, 5].

As a model of a uniprocessor real-time system, a single-channel queuing system with a heterogeneous flow of applications is widely used, which we will henceforth be called the base model [3], bearing in mind that other more complex models are based on it and can be calculated.

In the process of analyzing the quality of the functioning of the system, the following characteristics of the application service are calculated: waiting time; Time of application stay in the system; The number of applications in the queue and the number of applications in the system. The listed characteristics are defined for each class of applications, and for applications of the combined (total) flow.

For the base model, using the method of introducing an additional event [6] in the case of mixed priorities, Laplace transforms of the waiting time and stay of applications in the system are obtained, and the number of applications of different classes in the system that produce the mean values and the second moments (coefficients of variation) Maintenance of applications of each class and applications of the combined (total) flow.

Calculation of the characteristics of the basic model using discipline with static alternating priorities is performed at the level of two moments of the distribution of service characteristics of applications of each class in accordance with the methodology described in [4]. In this case, priorities can be assigned to application classes in any order.

For discipline with dynamically varying in time mixed priority in [7] recurrence relations are obtained for the average values of the service characteristics of applications of different classes in the case of a linear and nonlinear priority function.

Based on these results, approximate methods for calculating the service characteristics of applications were developed under the most general assumptions about the nature of the flows of applications and services [8]. Comparison with the results of simulation modeling demonstrates the satisfactory accuracy of the proposed approximate methods (the error does not exceed 10-20%).

Let's analyze, using the base model, the influence of various service disciplines on the performance characteristics of a single-processor system, which receives applications of four classes with intensities of 0.1; 0.2; 0.3; 0.4 [s^{-1}] and average service durations of 2.0; 1.0; 0.5; 0.1 [s], respectively. We will assume that the time intervals between incoming applications and the duration of their service are distributed according to an exponential law. As the main characteristic of the operation of the system, the waiting time is considered, on which the reaction time of the system directly depends.

The dependence of the average waiting time for applications of different classes when using no-priority and cycling priority with direct (1, 2, 3, 4) and reverse (4, 3, 2, 1) orders of priorities. As can be seen from the graphs given above, the waiting time increases with decreasing priority, and the waiting time conservation law formulated by L. Kleinrock [5] is fulfilled for an up-to-date state. The direct order of assigning priorities leads to a greater spread of waiting times for applications compared to the reverse order. This is due to different values of the intensity of arrival and the length of service of applications of different classes. The reverse order of prioritization allows to reduce the total average length of all queues of requests, in comparison with the absolute of the no-priority discipline and the direct order of assigning alternating priorities.

The use of absolute priorities and cycling priorities leads to an even wider spread in the waiting times for applications of different classes, with the greatest variance being achieved for absolute priorities with the reverse order of prioritization. Note that in the case of a direct order of prioritization, the waiting times differ from each other to a lesser degree than with the reverse order of priority assignment.

Comparison of the cycling and shows that the use of alternating priorities reduces the average waiting time for applications of the second, third and fourth classes due to a certain deterioration in the service of first-class applications. However, on the whole, the nature of the dependence for the cycling priority is similar to the nature of the dependencies for relative and absolute.

CONCLUSIONS

Analysis of the results, shows that the introduction of dynamic priorities allows to reduce the spread between the waiting times for applications of different classes in comparison with static priorities. The advantage of dynamic priorities in comparison with static ones is that, when moving from one strategy to another, a smooth (continuous) change in the waiting time for applications of different classes can be provided, while in a discipline with static priorities this change occurs in a spasmodical manner. However, disciplines with dynamic priorities are more complex in implementation than the same with static priorities. In single-processor, this results in a significant unproductive CPU time for dispatching. Therefore, the use of the discipline with dynamic priorities proves to be justified only when the parameters of the input streams of applications are sharply and often vary with time, and therefore, the discipline with static priorities cannot be used.

A comprehensive analysis of the ways to manage the flow of applications in the system with real-time shows the following:

- priority processing in a single mode, in contrast to the group mode, provides high-quality applications with fairly good quality of service (a slight increase in the waiting time) when the system is loaded close to unity due to a sharp increase in the waiting time for low-priority applications;
- introduction of dynamic priorities allows for a smoother change in the characteristics of data processing than in the case of static priorities, which allows more efficient provision of specified limits on the response time of the system;
- priority processing in group mode provides less unproductive expenses for dispatching in comparison with a single mode, which allows recommending group mode in systems with a high application rate ($10-100 s^{-1}$).

REFERENCES

- [1] Shwartz M. Computer networks: Analysis and development. — M. Radio and networking, 1981.
- [2] Majorov S. M. Computing systems basics. — M. High school, 1994.
- [3] Olifer V., Olifer N. Computer networks. SPb.: Piter, 2011. 944 pp.
- [4] Kupriyashkin A.G. Basics of system modeling. Norilsk: Norilsk industrial inst., 2015, 135 pp.

- [5] Kleinrock L. Queueing Systems: Volume I - Theory, New York: Wiley Interscience, 1975. - 417 pp.
- [6] Aliev T.I., Nikulsky I.Y., Pyattaev V.O. Modeling of packet switching network with relative prioritization for different traffic types // Proc.10th International Conference on Advanced Communication Technology, ICACT-2008. Phoenix Park, South Korea, 2008. Art. 4494220. P.2174-2176. doi: 1109/ICACT. 2008. 4494220
- [7] Aliev T.I., Muravieva-Vitkovskaya L.A. Priority strategies for traffic management in multi-service computer networks // Izv.vuz. Priborostroenie.2011, vol.54(6), pp. 44-48.
- [8] Aliev T.I., Muravieva-Vitkovskaya L.A., Sosnin V.V. Modeling: tasks, exercises, tests. SPb: NIU ITMO, 2011. 159 pp.