Router CPU Time Management using Particle Swarm Optimization in Cellular IP Networks
Mohammad Anbar, Deo Prakash Vidyarthi

Abstract

Providing QoS is one of the important issues in order to improve the performance of Cellular IP network. Resource reservation is often used in achieving this goal and it is proven to be an effective mechanism so far. Resources of Cellular IP network that can be reserved are bandwidth, buffer and CPU cycles. This paper is proposing a model for router CPU cycle reservation for real-time flows in Cellular IP network. The model applies Particle Swarm Optimization, as a soft computing tool, to reserve the CPU cycles and reduce the flow processing time at each router in the route taken by a flow. Simulation experiments reveal the efficacy of the model.

Keywords

Particle Swarm Optimization (PSO), Router CPU, Processing time, Packet Arrival Rate, Packet Processing Rate.

1. Introduction

As All-IP networks in which IP is used as a transport protocol are evolving rapidly, real-time applications such as video, data and voice need to be supported. Supporting real-time applications in IP networks involves improving the parameters affecting the QoS level in such networks. QoS parameters, in general, include delay, bandwidth, buffer, packets processing time at routers, jitter, etc. Improving the QoS in any network is the main concern for many researchers and substantive attempts have been made to improve QoS especially in All-IP networks. Nonetheless, the problem still gives the possibility of further improvement in parameters meeting QoS.

Resource reservation is one of the methods proven to be successful in improving QoS. Resource reservation includes the most important resources in the network such as the bandwidth [1] and buffer [2]. Resource Reservation Protocol (RSVP) is a common protocol for resource reservation [3]. As RSVP depends on signaling (messaging) as the main principle, it may flood the network with signals. This results in an extra load in the network and therefore wastage of the resources. Having such difficulties, RSVP, can be replaced by other solutions for resource reservation.

Soft computing can be of good help in optimization problems that are not having straightforward solutions [4]. Evolutionary Algorithms e.g. Genetic Algorithms, Particle Swarm Optimization, Support Vector Machine, etc. have been applied very successfully for many optimization problems. It is the powerful tool used quite often in resource reservation/ resource management as well [5].

In communication networks, CPU router cycle is an important resource, especially in Cellular IP networks, due to the wireless nature of these networks. CPU at router in Cellular IP networks takes some amount of time to process a packet. This time is attributes to the time delay during routing of a flow of packets from a source to a destination. Any router in Cellular IP networks (including base stations as routers) can process a certain number of packets per second. As a rare resource [6], router’s CPU cycle can be considered a precious resource to be reserved. When such a router’s CPU is loaded according to its admission control policy, it must consider real-time packets with higher priority than the non real-time packets. In other words, the router CPU must be reserved for real-time packet processing. The aim of this work is to reduce the processing time at each router on the flow path to the final destination. This applies PSO, as one of the efficient techniques for such problems, in optimizing the CPU cycle at each routers falling in the path of a flow. Experiments reveal the performance of the model.

The rest of the paper is organized as follows. Section 2 explains the routing operation in Cellular IP networks. The proposed model using Particle Swarm Optimization has been explained in section 3. The step-by-step algorithm, used in the model, is listed in section 4. Simulation experiments, to depict the performance of the proposed model, have been
2. Routing in Cellular IP networks

Cellular IP network is divided into cells. These cells are controlled by the base stations considered as access points and wireless routers in Cellular IP networks. Gateway connects the network to the Internet. Cellular IP protocol is designed for mobility management at the micro level. Mobile hosts in Cellular IP networks implement Cellular IP protocol. Many operations take place in this network such as handoff, paging and routing. Structure of Cellular IP networks is shown in figure 1.

Routing in Cellular IP networks is one of the important operations by which the packets of a flow are forwarded from source to destination. Two types of routing are often used in communication systems; end-to-end routing and hop-by-hop routing. In general, hop-by-hop routing is used in mobile communications [7] especially in Cellular IP [8]. Hop-by-hop routing may help in reducing energy consumption and packet processing time [9]. The main principle of hop-by-hop routing is that the packets are forwarded (routed) according to an independent decision taken at the router (base station in case of Cellular IP networks) based on the destination addresses for the incoming flow. Each base station, in Cellular IP networks, maintains a routing cache. Two types of information are stored in the routing cache; the source IP address and the previous neighbor from which the packet reached the current base station. Hop-by-hop routing differs from other types of routing by way of routing tables. In hop-by-hop routing one packet carrying the path to the destination information is enough as entry to the routing table and there is no need for carrying the full header to the destination by the packets to be followed [7]. Definitely, the route information must be updated through the data packets being sent. As long as the mobile host is sending packets through this route regularly, the routing cache will keep valid routing information. It is to be observed that route in Cellular IP network stays valid for a specific period of time known as route-time-out [8].

Each router in the Cellular IP network has a packet processing capacity i.e. each router has a policy to accept or reject a flow if the load exceeds its capacity. Any router cannot accept any flow if it is already loaded by the maximum allowed limit as per its policy (capacity) [6]. According to the queuing system theory, a flow must reserve enough router CPU cycles in order to avoid a long queue delay. Real-time flows, as delay sensitive flows, must be provided with the required CPU cycles to ensure that they are processed in a minimum time at the router.

Therefore, admission control, subjected to the wireless router capacity (policy) and minimizing the packets processing time at the wireless routers in Cellular IP networks can ensure better QoS in this network in sense of time delay for the real-time flows.

3. Router’s CPU time Optimization using PSO

Swarm intelligence is an intelligent paradigm to solve the hard problems. It is based on the behavior of the social insects such as bird flocks, fish school, ant colony etc. in which individual species change its position and velocity depending on its neighbor. This is done by mimicking the behavior of the creatures within their swarms or colonies.

Particle Swarm Optimization (PSO) is an algorithm based on the swarm intelligence. It is a population based tool used to find a solution to hard optimization problems. The problem to be solved (e.g. bandwidth reservation problem) can be transformed to the function optimization problem. As an algorithm, the main strength of the PSO is its fast convergence. Due to its well organized logic and procedures the optimal solution for a specific problem can be attained very fast. PSO shares many common points with Genetic Algorithm (GA). Both the algorithms start with a group of randomly generated populations and both have fitness function to evaluate the individual (solution) in the population. Also, both update the population (search space) and search for the near-optimal solution with random techniques [10].

PSO model is a swarm of individuals called particles. Particles are initialized with the random solutions. These particles move through many
iterations to search a new and better solution for the problem. Each particle is represented by the two factors: one the position, where each particle has a specific position and at the beginning initialized by the initial position \( x \) and the other factor is the velocity \( v \), where each particle moves in the space according to this velocity. During the iteration time \( t \), the particles update their position \( x \), and their velocity \( v \) [10].

PSO simulates the behavior of the bird flocking. Consider the following scenario. A flock of birds is randomly flying searching for the food in an area and there is only one piece of food in the area being searched. All the birds in the flock have learned that there is food in this area but none of them knows where the food is. The best strategy to locate the food is to follow the nearest bird to the food [11].

In PSO algorithm, there are two types of best values: one is \( P_{best} \), which is the best position for each particle in the swarm and must be updated depending on the fitness value for each particle. The second best value is \( G_{best} \), which is the global best value for the swarm in general. This value must be checked, and is exchanged by the best \( P_{best} \) if the \( P_{best} \) in this iteration is better than \( G_{best} \) for the last iteration.

The pseudo-code of the PSO algorithm is as follows [12].

```plaintext
PSO ( )
{
  Initialize the swarm by giving initial and random values to each particle.
  For each particle do
  {Calculate the fitness function
   If the value of the fitness function for each particle at the current position is better than the fitness value at \( P_{best} \) then, set the current value as the new \( P_{best} \).
   Choose the particle with the best fitness value of all the particles as the \( G_{best} \).
   Update the velocity of each particle as
   \[
   v_{j}^{k+1} = w \cdot v_{j}^{k} + c1 \cdot r1 \cdot (P_{best}^{k} - X_{j}^{k}) + c2 \cdot r2 \cdot (G_{best}^{k} - X_{j}^{k})
   \]
   Update the position of each particle as
   \[
   X_{j}^{k+1} = X_{j}^{k} + v_{j}^{k} \cdot \Delta t
   \]
  } Until the solution converges
}
```

In the pseudo code above, \( v_{j}^{k} \) is the velocity of particle \( j \) in iteration \( k \).

\( P_{best} \) is the best achieved solution for each individual so far.

\( G_{best} \) is the global best value for the swarm.

\( X_{j}^{k} \) is the current position of particle \( j \) in iteration \( k \).

\( w \) is inertia weight and is varied from 0.9 till 0.4.

\( r1, r2 \) are random numbers between 0 and 1.

\( c1, c2 \) are acceleration factors that determine the relative pull for each particle toward \( P_{best} \) and \( G_{best} \) and usually \( c1, c2 = 2 \).

\( \Delta t \) is the time step and usually 1.

Packet processing at a router starts when a packet reaches to this router. The time taken by the router to complete processing of a packet is \( T_{i} \). Therefore the time taken to process a flow consists of \( (N) \) packets is \( (N \times T_{i}) \). Total time taken to process a complete flow in the route, having \( M \) routers, is:

\[
T = \sum_{i=1}^{M} N \times T_{i}
\]  

(3)

Few assumptions have been laid down in this model and are listed as below.

Flow rather than a packet is studied in this model; therefore, the flow is assumed to be consisting of \( N \) packets.

Flows, in the model, are considered to be real-time flows and for this reason the procedure of admission control is not discussed. It is so as all real time flows are assumed to be accepted by the routers. The main concern of this model is the processing time at the router for each flow.

Each wireless router (base station) in the Cellular IP network has its own capacity (packet processing rate) referred as \( \mu_{i} \). Packet arrival rate at each router is different and is referred as \( \lambda_{i} \).

Queue is built up until the processing time of each flow consisting of \( (N) \) packets is completed.

Based on the queuing system theory the time taken to process a packet by a router [13] is:

\[
T_{j} = \frac{1}{\mu_{j} - \lambda_{j}} \times \frac{1}{\mu_{j}}
\]

(4)

The model considers routing operation in Cellular IP networks. In figure 1 an example of routing operation is depicted, where a flow (group of packets) passes its way (route) from the moment it enters the network through the Gateway which connects the network to Internet until it reaches the final destination. Each base station forwards the flow based on an independent decision, the main principle of Hop-by-Hop routing as discussed earlier. At each router (base
station) CPU takes time to process the packets and forward them to the next hop. This model reduces the total processing time taken on this specific route to process the flow. To achieve this objective the model applies Particle Swarm Optimization. Each router will be represented by its CPU processing time because this is the parameter to be optimized in this model. Two main parameters that affect the processing time at each router (evident from Eq. (2)) are processing rate of the routers and packet arrival rate. Due to these two parameters, it results in different processing times at the routers. Therefore, the total time taken to process a flow on a route is different for every new flow. To optimize the processing time at the routers of the flow becomes an optimization problem, which is desired to be as low as possible. PSO is applied as one of the powerful tested optimization tools to solve such optimization problems. PSO, in this model, minimizes the processing time taken to process a flow that passes through number of routers in the way.

As each individual of the swarm in PSO is represented by the two parameters (velocity and position), each router is considered as an individual in the swarm consisting of many routers (individuals). Therefore, at an individual in the swarm (router) two main parameters are considered; packets arrival rate ($\lambda_i$) and packet processing rate ($\mu_i$). Packet arrival rate is the velocity of each individual in the swarm; packet processing rate is the position of each individual in the swarm.

4. Algorithm used for the model

The GA based algorithm for the router’s CPU time minimization is listed as below.

1) Enter the number of routers (hops).
2) Enter the number of packets in the current flow.
3) Enter number of iterations.
4) Consider that packet arrival rate and packet processing rate values in different ranges of values for each experiment.
5) Initialize the swarm consisting of number of routers (individuals) by randomly generating packet arrival rate (velocity) and packet processing rate (position) for each router within the given range and computing the initial processing time at each router (individual).
6) Initialize the best processing time for each router (Pbest) by selecting the initial computed values as the best values for each router.
7) Initialize the global best value among the individuals in the swarm (Gbest) by selecting it as the smallest processing time among all the individuals in the swarm.
8) Compute the (initial) total processing time taken on the route having the entered number of routers. Total processing time on the route is the fitness function for the model.
9) For the entered number of iterations repeat the steps for 10 until 16.
10) Update the velocity of each individual (packet arrival rate $\lambda_i$) as in Eq. (1).
11) Update the position of each individual (packet processing rate $\mu_i$) as in Eq. (2).
12) Check the values of $\lambda_i$ and $\mu_i$, make sure that each value is within its allowed range.
13) Update (Pbest) for each individual as follows: if packet processing time for this router at iteration k+1 is less than packet processing time for the same router at iteration k then make Pbest as the packet processing time at iteration k+1.
14) Update (Gbest) as flows: for all routers if packet processing time of any of them at iteration k+1 is less than Gbest then make Gbest as the packet processing time for this individual (Gbest is the best value for the swarm found so far).
15) Compute the total processing time for the flow on the route.
16) Output the total processing time for the flow on the route consisting of entered number of routers.
17) Change the range of packet arrival rate and packet processing rate and repeat the whole algorithm all over again.

When a flow consisting of N packets enters the network through the gateway routed to a correspondent node in the network, it has to pass through many routers on hop-by-hop basis until it reaches to the destination. The number of routers, which a flow will pass through, is known in advance and therefore entering number of routers (hops) will be done at the beginning of the algorithm. The experiment using this algorithm is repeated for various numbers of routers 8, 12, 16 and 20 every time. As each flow has different number of packets, the algorithm asks for the number of packets per flow as an input also. Swarm initialization is done by randomly generating the values of packet arrival rate and packet processing rate in different ranges of values in order to check all possibilities and cases of packet processing times. As the objective of the algorithm is to minimize the total processing time in the route; the fitness function considers the total processing time at the entire route consisting of M routers. Two effective parameters ($\lambda_i, \mu_i$) controls the swarm in order to achieve the
best (smallest) packet processing time along the route. The individuals can move according to this algorithm within specific search area (range) which gives the limits of packet arrival rate and packet processing rate. After updating the values of $\lambda_i$ and $\mu_i$; they must be checked carefully not to exceed the allowed search space (range).

5. Simulation Experiments

Experiments are conducted with different values of packet processing rates and different values of packet arrival rates with the objective of minimizing the total processing time taken to process a flow consisting of $N$ packets. The experiments have been divided according to the values of packet processing rates and packet arrival rates as follows. Packet processing time is measured in (msec).

5.1 First set of experiment considers various processing rate of the routers and packet arrival rate at the router. Various graphs below depict the performance for different number of packets per flow with varying number of routers. For all the experiment in this set the input are as follows. Packet arrival rate is between 500 to 1000 packets/sec., Packet processing rate is between 1000 to 1500 packets/sec, and numbers of routers are 8, 12, 16, and 20 for four experiments.

5.1.1 Number of packets per flow is 200

5.1.2 Number of packets per flow is 400

5.1.3 Number of packets per flow is 600

5.1.4 Number of packets per flow is 800

5.1.5 The graph below shows the final results of four experiments (5.1.1-5.1.4) for various flows.
5.2 In this experiment, the packet arrival and packet processing rates have been changed. Packet arrival rate is between 1000 to 1500 packet/sec, Packet processing rate is between 1500 to 2000 packets/sec, Number of routers are 8, 12, 16, 20 for four experiments and again the number of packets per flow are 200, 400, 600, 800 for four experiments. In the graph only the final results are shown. In this experiment, the packet arrival and packet processing rates have been changed. Packet arrival rate is between 1000 to 1500 packet/sec, Packet processing rate is between 1500 to 2000 packets/sec, Number of routers are 8, 12, 16, 20 for four experiments and again the number of packets per flow are 200, 400, 600, 800 for four experiments. In the graph only the final results are shown.

5.3 Again, a different packet arrival and packet processing rates have been considered. Packet arrival rate is between 1500 to 2000 packet/sec, Packet processing rate is between 2000 to 2500 packets/sec, Number of routers are 8, 12, 16, 20 for four experiments, Number of packets per flow are 200, 400, 600, 800 for four experiments. The final results are shown in figure 8.

5.4 Other values of packet arrival rate and packet processing rates have been experimented with Packet arrival rate is between 2000 to 2500 packet/sec, Packet processing rate is between 2500 to 3000 packets/sec, and numbers of routers are 8, 12, 16, 20 for four experiments, Number of packets per flow: 200, 400, 600, 800 for four experiments.

5.5 Another experiment has been conducted to observe the processing time for various ranges of packet processing rate and packet arrival rate with different number of packets but fixed number of routers. The figure below depicts the result.
6. Observations and Conclusions

1) As the packet arrival and packet processing rates have been randomly generated, the first values of processing time are not proportional with number of routers. For example, in Fig 2 processing time for the flow when there are 20 routers is less than the processing time for the same flow when there are 16 routers. After applying PSO the difference is noticeable and the end value for processing time when there are 20 routers is higher than the same value when there are 16 routers.

2) Same behavior of the model is noticed when number of packets per flow increased up to 400 packets per flow as shown in Fig 3. The difference is in the end values of processing time and this is logical because increasing the number of packets per flow will lead to the increase in processing time.

3) The performance and the benefit of the model are clearer when the number of packets per flow is 800 as shown in Fig 5. After applying PSO the processing time is drops from 90000 msec. to 10000 msec.

4) The observation that can be made from Fig 6 is that number of packets per flow and number of routers on the route are the two affecting parameters that make processing time bigger when their values are big.

5) By increasing the ranges of values for packet arrival and packet processing rates the processing time is getting lower than the small ranges. (Fig 7, 8, 9)

6) Fig 10 shows a summary of all the conducted experiments. In this figure the processing time is getting smaller by increasing the ranges of values after applying PSO. The justification for that is as follows: the two parameters which are playing the direct and main role in changing the processing time according to Eq. (4) are packet arrival rate and packet processing rate and these two parameters are controlled during the experiment in producing better (smaller ) processing time. This justification can be proven mathematically by changing the Eq. (4) to be in this form:

$$T_i = \frac{1}{\mu_i - \lambda_i}$$  \hspace{1cm} (5)

From Eq. (5) it is clear that when the difference between ( $\mu_i$ and $\lambda_i$ ) are bigger then the value of processing time is smaller.

7) As a conclusion from the conducted experiments: the model can find out the best values for ( $\mu_i$ and $\lambda_i$ ) when number of routers is big for which the processing time is minimum.

7. References


