Research on Distributed QoS Routing Algorithm Fallback++ Based on Transaction and Multiobjective Programming

Feng Jie, Deng Jin Yang*, Wang Qin
Dalian Naval Academy, Dalian 116018, China
*fd8821@163.com

Keywords: Transaction classification; Distributed routing Algorithm; QoS; Time complexity; Simulation

Abstract. Transaction classification and QoS parameter Constraints are first researched. Then distributed QoS routing algorithm Fallback++ based on transaction and multiobjective programming is designed. Finally the time complexity and space complexity of the algorithm are analyzed. Simulation results show that Fallback++ is more efficient than Fallback and Fallback+.

1 Introduction

With the developing of computer network and communication technology the realtime media services of the TV meeting, VoD etc. are speedly advanced. In which the Quality of Service (QoS) is required because the information of image and sound is needed to be made communication realtime and stability. Therefore, multiple QoS routing algorithm is required to be researched which can satisfy the bandwidth, time delay and package lost rate etc. but it is a NP complete problem [1]. Many researchers put forward algorithms to solve the problem, such as [2,3,4]. In the shortest routing Fallback[4] there are following problems through analysis. (1) Although QoS parameters are simply classified they are not further divided, whose operation is not good. (2) When two or more paths are existed no way the optimal path is chosen without defining the standard of selecting the path. (3) The time complexity and space complexity is a little high. Facing to the above problems of the algorithm Fallback+, a distributed QoS routing algorithm Fallback++ based on transaction and multiobjective programming is put forward in this paper. The multiobjective programming models are respectively establishe through various transactions required and the prior order is determined according to satisfying QoS parameter constraints, which makes algorithm’s operation improve. The network utilization rate is defined as the standard of selection of the optimal path, when two or more paths exist, the optimal path can be chosen. The time complexity and space complexity of algorithm’s are lower. Finally, the correctness and efficiency of algorithm’s are verified by the simulation.

2 Transaction classification and QoS parameter Constraints

Four categories A、B、C、D are classified in the transactions of Internet whose requirement to QoS is different. According to Transaction classification and QoS parameter Constraints, the required QoS parameters are satisfied with the following conditions.

(a) Transaction A: A kind of the transaction A is nonrealtime data whose most part of communication at present in Internet belongs to A, such as E-MAIL, file transfer (FTP) and remote login (Telnet) etc.. the QoS required of transaction A is simpler, which is only to make package lost rate $L_r$ of QoS parameter smaller. (1) $f_x = 1 - \sum_{k=1}^{m} (1 - \sum_{j=1}^{n} x_{kj}) \times L_r$ (k=1,2,...m)

(b) Transaction B: A kind of the transaction B is image data which is newly developing transaction, such as noninteractive media communication, network TV, video point and broadcast etc.. Transaction A is strictly requiered of QoS parameters and which are needed to make band breadth $B_d$, time delay $T_d$ and package lost rate $L_r$ constraints which are prior order of QoS parameters.
(2) \[ f_1 = \sum_{j=1}^{\tilde{m}} b_j x_j < B_d \quad , \quad (3) \quad f_1 = \sum_{i=1}^{\tilde{m}} \sum_{j=1}^{\tilde{m}} d_{ij} x_{ij} < T_d \quad , \quad f_2 = 1 - \prod_{i=1}^{\tilde{m}} (1 - \sum_{j=1}^{\tilde{m}} f_j x_j) < L, \quad (k = 1, 2, \ldots, m) .

(c) Transaction C: A kind of the transaction C is realtime sound data which belongs to noninteractive media communication. Therefore their QoS parameter constraints and prior order are the same as the transaction B.
(d) Transaction D: A kind of the transaction C is video meeting data which is interactive media communication. To continuously media transfer, end-end time delay \( T_d \) and time delay jitter \( T_j \) are two key parameters. The application of interactive media communication is strict limit for time delay, otherwise, QoS is affected heavily by the parameter. Meanwhile, time delay jitter \( T_j \) is also strict limit, otherwise, which affects heavily person to identify for sound and image, and band breadth \( B_d \) is some required.

If two or more pathes are obtained through above QoS parameter constraints, the following network utilization rate \( f_7 \) is used to select the optimal path: \( (5) \quad f_7(x) = \sum_{i=1}^{\tilde{m}} \sum_{j=1}^{\tilde{m}} e_i x_{ij} x_j \times 100\%

3. Algorithm Fallback++ and its complexities
3.1 Algorithm Fallback++
The basic thought of algorithm Fallback++ is added to generate the optimal path according to the transaction classification in Fallback, when two or more pathes exist, The network utilization rate is defined as the standard of selection of the optimal path, whose steps is as follows.

Input: Source \( s \), target \( t \), adjecet matrix \( w(u, v) \), QoS parameter constraints.
Output: The path \( P \) from Source \( s \) to target \( t \) satisfying with QoS parameter constraints.

Step1: Run algorithm Dijkstra and generate all the shortest path set \( P_i, i = 1, 2, \ldots, k \) and store them in the stack.
Step2: The \( f_1, f_2, f_3, f_4, f_7 \) of all pathes are calculated respectively accoeding to formula (1)–(4) and (5).
Step3: Excute the following statement according to different transactions.
Case A If \( f_2 > L_r \) then delete the path \( P_j, j = 1, 2, \ldots, l, l < k \) that doesn’t satisfy QoS parameter constraints which forms a probable path set \( P_p = P_i - P_j \), \( p \leq k \).
Case B If \( (f_1 < B_d \quad OR \quad f_3 > T_d \quad OR \quad f_2 > L_r) \) then delete the path \( P_j, j = 1, 2, \ldots, l, l < k \) that doesn’t satisfy QoS parameter constraints which forms a probable path set \( P_p = P_i - P_j \), \( p \leq k \).
Case C If \( (f_1 < B_d \quad OR \quad f_3 > T_d \quad OR \quad f_2 > L_r) \) then delete the path \( P_j, j = 1, 2, \ldots, l, l < k \) that doesn’t satisfy QoS parameter constraints which forms a probable path set \( P_p = P_i - P_j \), \( p \leq k \).
Case D If \( (f_3 > T_d \quad OR \quad f_4 > T_j \quad OR \quad f_1 < B_d) \) then delete the path \( P_j, j = 1, 2, \ldots, l, l < k \) that doesn’t satisfy QoS parameter constraints which forms a probable path set \( P_p = P_i - P_j \), \( p \leq k \).
Step4: Excute the following statement to a probable path set \( P_p \) in the stack.
Case B If \((d_1 > B_d \text{ and } f_3 < T_d \text{ and } f_2 < L_r)\) then take the path that satisfies QoS parameter constraints from the stack.

Case C If \((f_i > B_d \text{ and } f_3 < T_d \text{ and } f_2 < L_r)\) then take the path that satisfies QoS parameter constraints from the stack.

Case D If \((f_3 < T_d \text{ and } f_4 < T_j \text{ and } f_i > B_d)\) then take the path that satisfies QoS parameter constraints from the stack.

Step 5: If only one path satisfies all QoS parameter constraints GOTO Step 7; otherwise, take every probable path from the stack and calculate network utilization rate \(f_7(P_p)\).

Step 6: \(P_p\) is selected from \(\max f_7(P_p)\) which is the optimal path \(P\).

Step 7: OUTPUT \((P)\);

Step 8: END.

If the minutest cost path is generated by use of algorithm Dijkstra but the selected path doesn’t exist which satisfies QoS parameter constraints, algorithm Fallback processes as follows. The greater weight coefficient QoS parameter constrain is select as the optimal objective function and the other QoS parameter constraints don’t considered, by which the optimal path is generated. In algorithm Fallback+ when the minutest cost path and the optimal path are generated the probable paths are stored in the stack. If the generated shortest path is not satisfied with QoS parameter constraints then search from stored paths and if the path satisfied with QoS parameter constraints then select the path. If all probable paths don’t satisfied with QoS parameter constraints then algorithm Fallback is used to process. But in algorithm Fallback++, QoS parameters and transactions are subtly divided and network utilization rate is selected as the optimal objective function. Through these operations, the paths to be chosen and the probability of selection of the path satisfied with QoS parameter constraints are increased. Meanwhile, the operation of algorithm’s is enhanced and the time complexity and space complexity of algorithm’s is decreased.

3.2 Time complexity

In typical algorithm Fallback, supposed that all nodes in network are \(N\) and the path is generated by algorithm Dijkstra. For every QoS parameter the time complexity is \(O(N \times \log N)\). If the number of QoS parameters is \(n\), the time complexity of algorithm Fallback’s satisfied with QoS parameter constraints is \(O(n \times N \times \log N)\). But in algorithm Fallback+, the time complexity of algorithm Fallback+’s satisfied with QoS parameter constraints is \(O(n \times N \times \log N) + O(n \times N^2) = O(n \times N^3)\), \(n \leq 6\). But in algorithm Fallback++, the time complexity of algorithm Fallback++’s satisfied with QoS parameter constraints is \(O(n \times N^2)\), \(n \leq 3\). Because the communication transaction in Internet is divided carefully the time complexity is decreased which can be verified in the following simulation.

3.3 Space complexity

In typical algorithm Fallback, supposed that all nodes in network are \(N\) and the path is generated by algorithm Dijkstra. For QoS parameter table, stored space and stored cost the space complexity is \(O(n \times N^2)\). But in algorithm Fallback+, the space complexity of algorithm Fallback+’s satisfied with QoS parameter constraints is \(O(n \times N^2)\), \(n \leq 6\). But in algorithm Fallback++, the space complexity of algorithm Fallback++’s satisfied with QoS parameter constraints is \(O(n \times N^2)\), \(n \leq 3\). Because the communication transaction in Internet is divided carefully the space complexity is decreased.

4 Simulation and its result analysis

4.1 Simulation network topology

To increase the comparision of the simulation result, the same network topology is adopted as reference [5], which is shown in Fig.1. the QoS parameters of network topology are generated by
the random method[7]. Every node has four average links and the network topologies of 100 nodes are 10 and the QoS parameters of each link are 3, such as the time delay, band breadth, and packet lost rate which randomly takes interger in [1,10]. The optimal path is the smallest cost path satisfied with QoS parameter constraints. The smallest cost is referenced to as the smallest linked nodes on the same path. According to above conditions the routing success rate, optimal routing selection rate and CPU processing time are measured on COMPAQ 6520 whose simulation result is shown in Table 1.

![Network topology](image)

**Figure 1. Network topology.**

### 4.2 Result analysis

Seen from table 1 of the Simulation results, in the routing success rate, Fallback++ improves 1% compared with Fallback+ and 4% compared with Fallback, in the optimal routing selection rate, Fallback++ improves 4% compared with Fallback+ and 10% compared with Fallback, in the CPU processing time, Fallback++ saves 0.01 ms compared with Fallback+ and 0.067 ms compared with Fallback.

### 5 Conclusions

Transaction classification and QoS parameter Constraints are first researched. Then distributed QoS routing algorithm Fallback++ based on transaction and multiobjective programming is desiged. Finnally, the time complexity and space complexity of the algorithm’s are analysed. Simulation results show that Fallback++ is more efficient than Fallback and Fallback+ in the routing success rate, the optimal routing selection rate and the CPU processing time.

### References