Quality of Service routing in mobile wireless networks

Gonzalo Camarillo
Advanced Signalling Research Laboratory
Ericsson, FIN-02420 Jorvas, Finland
Gonzalo.Camarillo@ericsson.com

Abstract

Wireless networks present some characteristics that make algorithms used for wired networks unsuitable. Algorithms used for providing QoS paths through ad hoc networks have to work with imprecise information. Saving battery power and keeping the overhead of the routing protocols are important design rules.

Flooding algorithms with modifications to limit the number of routing messages generated seem to be the best solution for QoS routing in ad hoc networks. Imprecise models for delay and bandwidth calculations are necessary to provide the proper QoS metrics to the algorithms.

1 Introduction

Wireless networks consist of a set of nodes that communicate without having cables between them. Typically radio over the air interface is used. Wireless networks differ substantially from wired ones. Bandwidth becomes much more expensive and it should be saved as much as possible. This implies that some protocols that are suitable for wired networks cannot be employed successfully in a wireless environment. Overhead has to be kept very low and several compression techniques are employed in order to reduce the number of bytes transmitted on the air.

When mobility is added to the picture, new requirements are imposed to the protocols on be used. The transmission power of the different nodes has to be controlled to avoid interference and to save battery power. Therefore, protocols in these environments have to carry this kind of information.

Nodes can leave and reenter the network at any moment. This does not have to affect the stability of the network and it should be possible to employ algorithms that work with imprecise information [1]. There are some moments when the nodes in the network do not have a clear picture of the status of all links and peer nodes.

The term mobile wireless network is still far too general. There are several types of networks that can be referred to as mobile wireless. Some networks make use of a wired infrastructure for communicating between different nodes. The mobile nodes are typically one hop away from the base station, which is fixed. Thus, the mobile node has just to discover the proper base station to communicate to. The base station will then perform all the necessary actions in order to reach the destination. This systems split the space into cells and one or several base stations give coverage to each cell. Cellular telephone systems belong to this group of mobile wireless networks.

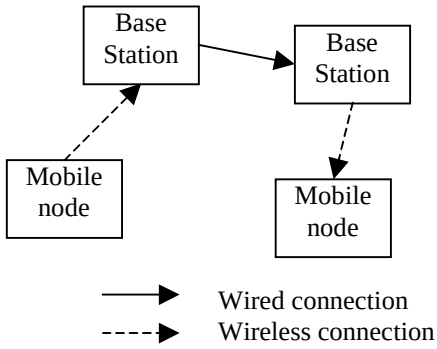

Figure 1 : Cellular system

In order to provide QoS in these networks it is necessary to provide the proper radio bearer and ensure that the handoff between base stations is smooth. The provision of QoS in the wired section can be undertaken using traditional QoS techniques that are appropriate for wired networks [2]. UMTS and GPRS systems, for instance, provide this QoS.

The UMTS mobile node negotiates the profile of the traffic to be transmitted with the network. then the proper radio bearer for this traffic is established. When the mobile gets into a different cell the network establishes another radio bearer with similar characteristics in the new cell. The result is a seamless handoff for the mobile.

There is another type of network where all the nodes move. These mobile nodes can act as source or destination nodes but also act as routers for another nodes' communications. Therefore, the packets sent from one host to another may traverse a set of mobile wireless hops before reaching their destination. These networks are typically known as Ad Hoc networks [3]. Ad hoc networks break in some sense the IP paradigm where there is a set of end points joined by a set of

routers in the middle that perform as efficiently as possible a simple and well defined task (routing). In Ad hoc networks the routers are also end points.

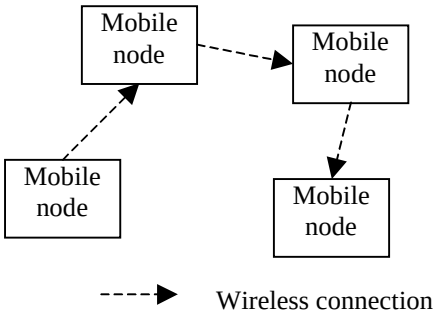

Figure 2 : Ad hoc network

There are also hybrid networks where the nodes are multiple hops away from the base stations. Ad hoc routing is used up to the base station.

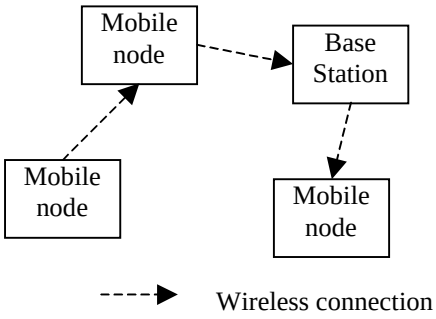

Figure 3 : Hybrid network

This paper focuses on ad hoc networks and how to provide QoS. There are already several solutions for routing packets in ad hoc networks on a best effort basis. Finding paths that fulfill certain requirements and are suitable for certain applications requires more advanced solutions.

It is worthwhile mentioning that QoS in ad hoc networks is always soft QoS. That is, no extreme requirements are imposed to the connections. Broken paths can appear and rerouting of packets may be necessary. For mitigating the effects of this, tailored algorithms are employed. For instance, voice or video can be transmitted using adaptive rate compression schemes.

Ad hoc networks can be utilized in disaster areas by the rescue teams, for military applications when soldiers do not count on a wired infrastructure, conference meetings etc…

2 QoS path discovery

The main purpose of QoS routing is to find a path that fulfills the QoS requirements of the connection requested. If this path does not exist, or the routing protocol used cannot find it, the connection requested will be refused. Therefore, some kind of admission control can be performed in the network with the information provided by routing protocols. Thus, the concepts “routing” and “resource reservation” are tightly related.

There are several ways to find a QoS path. In source routing every node maintains a image of the global network. This image of the network is used for choosing the appropriate path to route the packets. All the computation is performed in the source node.

Distributed routing performs a distributed computation. Nodes exchange control messages between them. This way the information store in every node is used for finding the path.

Hierarchical routing distributes the nodes into clusters [5]. Inside these clusters either source routing or distributed routing is used. There can be several levels in the hierarchy.

3 Source Routing

Source routing algorithms perform centralized path computation [6]. The source node calculates the path the packets will follow. Then the packets are sent through that path. Every packet carries a list of the nodes in the path. With this information the packets are routed towards the destination. Since packets carry the whole path nodes in the middle do not need to keep routing state. However, size of the packets increases when long paths are traversed.

The number of routing messages exchanged by source routing protocols is less than in distributed routing protocols. Source routing protocols are suitable for small networks. Networks where packets have to traverse many hops before reaching their destination impose high computational costs. Computing feasible paths in a centralized manner for a large network is very expensive.

When QoS is to be provided using source routing algorithms the information used for calculating the routes is often stale. Algorithms have to be prepared to work with imprecise information.

4 Distributed routing

Distributed routing uses information stored in the nodes for computing the proper path to the destination. It avoids centralized path computation. Centralized path computation can be very expensive for large networks. There are several algorithms that can be used for distributed path computation.

Flooding consists of trying all the possible routes from the source to the destination. It floods routing messages. The first message that arrives to the destination carries the path it has traversed. If the delay accumulated by this routing message is below the required delay the path can be used for the connection. This method almost always finds the best path since all the possible paths are tried. However it has also disadvantages. It overloads the network with multiple routing messages. In wireless networks where bandwidth is scarce this kind of algorithm consumes too much resources.

Other algorithms such as the shortest-path algorithm (SP) calculate just one path to the destination. If the calculated path breaks during the connection a new path has to be calculated. These algorithms have a small overhead since fewer messages are exchanged but do not provide reliability. They do not calculate secondary paths. In some situations when the information stored in the nodes is not precise enough, the calculated path is not the optimal.

Therefore, both approaches present advantages and disadvantages. An algorithm that combines good features from both approaches has to be employed. It is desirable that this algorithm does not overload the network with too many messages but still finds the optimal path providing secondary paths for reliability.

The TBD (Ticket-Based Probing) algorithm [3] attempts to combine all these features. It performs parallel path searches, but it limits the amount of routing messages generated. It also provides a means for calculating the expected bandwidth and delay of the paths. TBD takes also the cost of the links traversed into consideration.

4.1 Ticket-Based Probing algorithm

TBD algorithm follows the same approach as the flooding algorithm. TBD searches several paths in parallel and finally chooses the one that best suits the requested connection. However, TBD limits the number of searches performed. This way, the traffic generated by the routing algorithm is substantially decreased.

TBD uses three QoS metrics: delay, bandwidth and cost. The delay includes the radio propagation delay, the queuing delay and the protocol processing time in the path. The cost of the different links can be based on different factors.

The path delay is the sum of the delays of the links in the path. The path bandwidth is the bandwidth of the link with minimum bandwidth in the path. The cost is the sum of the costs of every link in the path.

In order to assign an appropriate cost to every link it is important to divide links into two classes: stationary links and transient links. There are several factors that influence this classification. Nodes that are moving are more likely to be moving at the next moment. A link that has just been established is not marked as stationary after a certain period of time.

Stationary links have to be used as much as possible in QoS paths. When a network contains basically transient links is very difficult to provide QoS since broken paths appear continuosly. QoS is provided in networks with a limited degree of mobility.

In order to make stationary links preferable for building QoS paths transient links are assigned higher costs than stationary links. This way, when several paths fulfill the delay or bandwidth requirements the one with lower cost will contain less transient links.

The information that the nodes have about the delay and the available bandwidth in the network is not accurate. The delay information coming in a refresh message can be obsolete when the refreshment time is comparable to the speed of the nodes. Therefore, algorithms dealing with path delays and bandwidths have to work with imprecise information. In wired networks the delay and bandwidth information received for a certain path is stored. A probability distribution is calculated for this path. Thus, the statistical model can predict the delay and the available bandwidth of the path at any moment. This method is not suitable for mobile wireless networks. Links can be short lived. They are not up for long enough to build a probability distribution. Therefore, a simple imprecision model has to be used.

The model for the delay calculation is explained below. Bandwidth calculations are performed in a similar manner.

A node gets the delay of the path (D^{new})every time a refresh message is received. The increment or decrement that the delay has suffered can be calculated by simply subtracting the new delay from the last value of the delay for the path ($\Delta D = D^{new} - D^{old}$). This delta value is stored. When a new refresh message with new delay information is received, this delta value is included in the formula ($\Delta D^{new} = \alpha \cdot \Delta D^{old} + (1-\alpha) \cdot |D^{new} - D^{old}|$, $\alpha<1$). This smoothes changes in the interval for the current delay of the path. α measures how fast or slow the

history of ΔD is forgotten. For $\alpha=0$ the previous values of ΔD are not taken into account. This algorithm is similar to the one TCP employs for calculating the RTT (Round Trip Time) of a path.

The current delay of the path is considered to be inside the interval ($D^{new} - \Delta D^{new}$, $D^{new} - \Delta D^{new}$). D^{new} is provided by a distance vector routing protocol.

Every node stores in its routing table D^{new} and ΔD^{new} for every possible destination. The first hop to the destination can be any of its neighbors. So, the size of the routing table depends on the total number of nodes and the number of nodes that are neighbors.

The table below shows part of the routing table of node i. In this example the node i has three neighbors: nodes 4, 5 and 6. The delay towards all the possible destinations following all different paths are shown.

Table 1 : Routing table example

Destination	**Next hop**	**D^{new}**	**ΔD^{new}**
Node 1	Node 4	220 ms	15 ms
Node 1	Node 5	300 ms	30 ms
Node 1	Node 6	100 ms	20 ms
Node 2	Node 4	130 ms	10 ms
Node 2	Node 5	115 ms	15 ms
Node 2	Node 6	200 ms	10 ms

Assuming that D^{new} and ΔD^{new} are stored using 4 bytes for each value and a network with 100 nodes, if node i has 10 neighbors the size of its routing table would be $10 \cdot 100 \cdot (4 \text{ bytes} + 4 \text{ bytes}) = 8000$ bytes. Note that this is a rough calculation. There is no need for keeping information about routes to itself, and just QoS information related to delay is taken into account in this calculation.

When a connection request arrives the source node sends one or several routing messages in order to find paths to the destination that fulfill the QoS requirements of the connection (delay and/or bandwidth). These routing messages, called probes, carry the accumulated delay for all the links traversed so far.

Logical tickets are used in order to limit the number of probes generated. Every probe contains a number of tickets. A probe must at least contain one ticket but can contain more. When a probe arrives to a node, several probes can be generated and sent to different next hops. Every probe will have a subset of the tickets that arrived in the original probe. Therefore, if a probe containing a single ticket arrives to a node it cannot be split into several ones. This limits the number of probes that can be present in the network, reducing the overhead generated by the flooding algorithm.

There are two kinds of tickets: green tickets and yellow tickets. Green tickets are used to maximize the probability of finding a low-cost path. Therefore, probes with green tickets are sent to the link with the lowest cost that still fulfills the delay requirements. Probes with yellow tickets are sent to the link with minimum path delay. Yellow tickets attempt to maximize the probability of finding a feasible path.

Probes arriving to the destination contain the path that they traversed and the accumulated delay. Based on this information resource reservation for a path can be performed.

This algorithm allows different level of reliability and redundancy. In the first level resources are reserved through several paths and data packets are sent through all of them. The destination simply drops the duplicate packets. In the second level reservations are performed through different paths, but one is marked as the primary path and the rest as secondary paths. Data packets are just sent though the primary path. If the primary path fails the traffic is diverted to one of the secondary path (if it is still valid).

Several simulations have been carried out comparing these algorithms. The flooding algorithm achieves the best success ratio. That is, it finds a feasible path for tough QoS requirements. TBD achieves very similar results, even with high imprecision rates. This shows that the paths discovered by TBD are usually the best ones. When the imprecision rate is high, the SP algorithm performs very badly. TBD achieves lower average path cost because both costs of the links and delay are taken into consideration.

5 Available bandwidth calculation

The previous section explains how bandwidth and delay information is used by the routing protocols. In wired networks the calculation of the available bandwidth in a path is straight forward. The path bandwidth is the bandwidth of the link with the least bandwidth. In ad hoc networks this calculation is much harder. There are several issues that have to be taken into consideration in the radio environment.

TDMA (Time Division Multiple Access) is usually employed in ad hoc networks. TDMA divides the time into slots. One or several slots are assigned for a connection. The figure below shows a very simple scenario, with just 3 nodes. Time slots 1, 2, 3 and 4 are available between A and B and between B and C. However B cannot send information to C in the time

slots that A is sending info. Since A, B and C are transmitting in the same frequency interference would occur if B tries to send data to C and receive data from A at the same time.

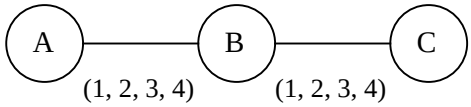

Figure 4 : TDMA system

Therefore, some time slots have to be assigned for the connection from A to B. Different time slots will be assigned for the leg from B to C. If B receives from A during time slots 1 and 2 and sends to C during time slots 3 and 4, the path bandwidth between A and C is 2 time slots.

CDMA (Code Division Multiple Access) allows to reuse the same time slot for different connections without interference. It uses orthogonal codes. The information is multiplied by a code and then it is decoded at the destination. This allows different nodes to transmit at the same time. Node 1 wants to send some data S1. Before sending it, it multiplies it by its CDMA code C1. The result is sent over the radio interface S1·C1. Node 2 performs the same operation with the data it wishes to send. S2·C2 is sent over the air. Both, S1·C1 and S2·C2 are sent at the same time, using the same time slot. The information on the air is S1·C1 + S2·C2.

A node interested in receiving the information sent by node 1 has to use C1 for extracting S1 from the radio interface. The received signal is multiplied by C1. Since the codes are chosen to have certain properties (C1·C1= 1 and C1·C2=0), the result at the destination is: (S1·C1 + S2·C2) ·C1 = S1.

In the simple example shown above the use of CDMA helps only if a mobile node is able to receive and transmit at the same time. However, in more complicated networks CDMA is very useful.

Ad hoc networks using TDMA and CDMA are divided into clusters. Inside the clusters TDMA is used, and CDMA allows to use the same time slots in different clusters. There are three types of nodes in this architecture. The cluster head keeps track of the available time slots in the cluster. All the time slots reservations are made based in the information provided by the cluster head. The cluster head can hear all the stations in the cluster. Gateway nodes belong to more than one clusters and perform the code swapping between different clusters. The rest of the nodes do not have any special function.

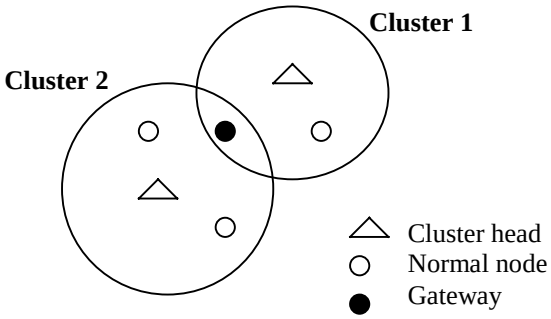

Figure 5 : Clustered system

The following example shows how the nodes get the bandwidth information for the path in a TDMA system using CDMA [7]. The figure below shows the configuration of the system.

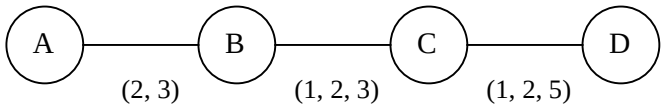

Figure 5 : Bandwidth computation

Every node knows the common free slots between itself and its neighbors. Periodic free_slots messages are transmitted for this purpose between adjacent nodes. Bandwidth messages are transmitted to calculate the path bandwidth to a certain destination.

In this example the path bandwidth from A to D is calculated. C knows the free slots between C and D because it has received a free_slots message from D. C sends a bandwidth message to B bandwidth (D: 1, 2, 5).

B has the following information:

- free slots between B and C (1, 2, 3)
- available slots between C and D (1, 2, 3)

There are three types of slots:
X= Slots that can only be used between B and C = 3
Y= Slots that can only be used between C and D = 5
Z= Slots that can be used on both paths = (1, 2)

The issue here is how to allocate Z between both paths. In this example the obvious choice is to assign 1 to CD and 2 to BC. Therefore, B sends to A bandwidth (D: 2, 3).

Node A can now calculate the path bandwidth to D. Slot 2 can be used in BD and slot 3 in AB. Therefore, the bandwidth AD is 1 time slot.

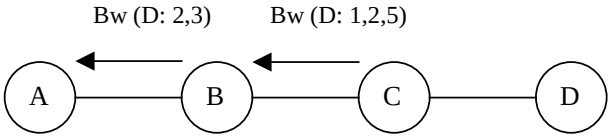

Figure 6 : Bandwidth messages exchanged

The distribution of Z between both links becomes an issue in larger systems. There are several proposed algorithms for undertaking this distribution. The simplest one is to assign half of Z to one link and the other half to the other link. More advanced algorithms take into account the size of X and Y. For instance, if X is large and Y has a small size more time slots from Z should be assigned to the link corresponding to Y. Taken this into consideration the path bandwidth is bigger. Thus, the number of connections refused decreases, and the success ratio grows.

6 Conclusions

QoS routing in ad hoc networks requires special algorithms to handle all the issues specific to wireless mobile systems. Flooding algorithms are the best for finding feasible paths with tough QoS requirements. They try every possible path to the destination. Therefore, the best path is almost always found. However, the amount of routing messages generated by these algorithms makes them unsuitable for systems with limited bandwidth.

Algorithms that use available information for finding the most suitable path do not work well in ad hoc networks. Information is often imprecise and the resulting path is not the best possible. The best solution for QoS routing in ad hoc networks comes out from a trade off between these two approaches. The TBD algorithm uses a flooding approach limiting the number of messages in the network. Thus, it combines good features from both types of algorithms.

Source routing algorithms are suitable for small ad hoc networks. When the size of the network grows centralized path computation becomes too expensive. Distributed routing algorithms suit large ad hoc networks requirements better.

Special algorithms are used to calculate the path bandwidth and path delay in ad hoc networks. Algorithms dealing with delays have to work with imprecise information. Algorithms dealing with bandwidth take into consideration TDMA slots and assignment of common slots for different legs of the connections.

7 Acronyms

CDMA: Code Division Multiple Access
GPRS: General Packet Radio Service
IP: Internet Protocol
QoS: Quality of Service
RTT: Round Trip Time
SP: Shortest Path
TBD: Ticket-Based Probing
TDMA: Time Division Multiple Access
UMTS: Universal Mobile Telecommunications System

References

[1] Guerin R., Orda A., “QoS-based routing in networks with inaccurate information: Theory and algorithms”, in Proc. IEEE INFOCOM’97, Japan, pp. 75-83.

[2] Chen S.,Nahrstedt K., “An overview of quality-of-service routing for the next generation high-speed networks: Problems and solutions”, IEEE Network, Special issue on Transmission and Distribution of Digital Video, pp. 64-79. Nov/Dec 1998.

[3] Chunhung Richard L., “QoS Routing in Ad Hoc Wireless Networks”, IEEE, 1998.

[4] Chen S., Nahrstedt, “Distributed Quality-of-Service Routing in Ad Hoc Networks”, IEEE journal on selected areas in communications, vol 17, no 8. August 1999.

[5] Ching Hsu Y., Tzu-Chieh T., Ying-Dar L., “QoS Routing in Multihop Packet Radio Environment”, IEEE. 1998.

[6] Broch J., Johnson D., Maltz A., “The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks”, draft-ietf-manet-dsr-03.txt, work in progress. October 1999.

[7] Chen T., Tsai J., Gerla M., “QoS Routing Performance in Multihop, Multimedia, Wireless Networks”, IEEE. 1997.