Multicast VoIP traffic in Ad Hoc Networks

JIMMI GRÖNKVIST

FOI-R--2750--SE
ISSN 1650-1942

Technical Report
April 2009

Information Systems

FOI
Swedish Defence Research Agency
Information Systems
P.O. Box 1165
SE-581 11 LINKÖPING

Phone: +46 13 37 80 00
Fax: +46 13 37 81 00

www.foi.se

Jimmi Grönkvist

Multicast VoIP traffic in Ad Hoc Networks

Title	Multicast VoIP traffic in Ad Hoc Networks
Titel	Multicast VoIP trafik i Ad Hoc-Nät
Report No. / Rapportnr	FOI-R–2750–SE
Report Type	Technical Report
Rapporttyp	Teknisk rapport
Month / Månad	April / April
Year / Utgivningsår	2009
Pages / Antal sidor	14
ISSN	1650-1942
Customer / Kund	FMV
Kompetenskloss	22 Samband och Telekommunikation
Project No. / Projektnr	E73502
Approved by / Godkänd av	Anders Törne

FOI, Swedish Defence Research Agency	FOI, Totalförsvarets Forskningsinstitut
Information Systems	Avdelningen för Informationssystem
P.O. Box 1165	Box 1165
SE-581 11 LINKÖPING	581 11 LINKÖPING

Abstract

In this report we study the possible performance of multicast voice type traffic in a multihop ad hoc network, more specifically we assume the medium access control to be TDMA-based. We primarily give a delay estimation, but capacity will be discussed as well.
We show that it may very well be possible to handle voice traffic over multiple hops given that the transmission schedule is adapted for it. Time slot length is particularly a very important parameter, and needs to be kept short if voice traffic is to be successful. We also shown that the link capacity rapidly is becoming a limiting factor for larger networks.

Keywords: Ad hoc networks, VoIP, CDS

Sammanfattning

I den här rapporten studerar vi hur väl multicast rösttrafik kan hanteras i ett multihopp ad hoc-nät, speciellt ett TDMA-baserat sådant. Vi kommer i första hand undersöka detta ur fördröjningssynpunkt, men även kapacitetskrav kommer att studeras.
Vi visar att det kan kan vara mycket möjligt att hantera multicast rösttrafik över multipla hopp, givet att vågform och sändningsscheman anpassas efter trafikkraven. Tidluckelängden är en speciellt viktig parameter, som måste hållas så kort som möjligt för möjliggöra detta. Vi visar också att för större nät börjar länkdatatakten bli en begränsande faktor.

Nyckelord: ad hoc-nät, VoIP, CDS

1 Introduction

In this report we study the possible performance of voice type traffic in a multihop ad hoc network, more specifically we assume the medium access control to be TDMA-based. We primarily give a delay estimation for voice sessions, although capacity will briefly be discussed as well.

Our primary focus is point-to multi-point voice sessions and the solutions given will be chosen with this assumption in mind. The solution will work also for point-to-point sessions, but better solutions can probably be designed. We assume that normally only one node at a time is generating packets at a time in a voice session; although any of the nodes may switch to become a source, i.e. start talking.

The interaction between the routing and medium access control is important when dealing with voice sessions, this is especially the case when we are dealing with TDMA-based access. Assuming a network with fairly low load, which can be seen as relatively realistic for voice traffic (other traffic probably must be given lower priority unless its delay requirements are even more severe than that of voice), a large proportion of the total delay will be spent waiting on next time slot. Since this is the case for each hop, it is important to decrease the perceived frame length for each hop. One of the ways to do this is to reduce the number of nodes retransmitting the packet on the way from source to all destinations (thus allowing more time slots per time unit to the remaining nodes).

This is a well studied problem in broadcast routing, but some things should be noticed. All nodes that do retransmit packets need a timeslot to do this. Every time the retransmitting nodes change we will have a change in the time slot structure, such change takes time to handle and may potentially be costly in terms of overhead. Thus, as long as the present number of transmitting nodes have a sufficient number of time slots to fulfil delay guarantees it may be better to keep the network as it is, even if better solutions exists (e.g. fewer nodes can be used to reach the network). Furthermore, we want the impact of switching speaker to have as little impact as possible; source node will of course change, but preferably as few retransmitting nodes as possible.

If we use a different set of retransmitting nodes depending on source, this means that we either need to assign timeslots to all nodes that are used for retransmitting for any possible source, or we need to handle time slot reassignments every time a speech session switches speaker (beside the source).

An additional advantage with using an equal setup is the case with multiple simultaneous sessions (with different sources), since this means that all retransmitting nodes will carry an equal traffic load, rather than having different traffic loads on the retransmitting nodes.

A common tree structure with these properties is a Connecting Dominating Set (CDS) [1, 2] and the nodes that are included into this set are called CDS nodes, i.e. what we above have denoted as the retransmitting nodes. Different techniques exist to create and update CDS trees, e.g. multi point relays can be used [3], but this will not be the main focus of this report. It should be noted though that all methods for creating CDS trees may not be useful for our purposes. A good CDS algorithm here should be locally updateable, change the previous set as little as possible and if possible be done simultaneously with the time slot scheduling updates. The last part has normally not been the focus of the CDS research, and further work may be needed to reduce the update time.

The source node is normally not included in the CDS tree and will probably be dealt with specifically. It will only change when we change speaker. Several methods can deal with this problem. Initial access can (to reduce response time) be handled though random access, but after this the node needs a more permanent time slot (or at least something remaining the length of the users talk).

2 Assumptions and Notations

We start with these basic assumptions:

- Time slotted schedule with equal size time slots.
- The CDS tree is equal for all source nodes which are always assumed to be given an equal amount of time slots (>0).
- The rest of the nodes are either only given time slots if they are the source of a voice session or all nodes are given one time slot each. In either case, nodes not in the CDS but with assigned time slots are called assigned nodes. We will seperate these cases with case A and B whenever necessary.
- The schedule can logically be divided into two parts, one for CDS nodes and one for assigned nodes. X parts of the time slots are given to the CDS nodes and Y parts of the time slots are given to the assigned nodes.
- Administrative slots, e.g. bootstrap slots, are given Z parts of the time slots
- $X + Y + Z = 1$

- All time slots given to a node are perfectly spread, i.e. if a node has four time slots per second there will be 250 ms between these time slots.
- We ignore effects of spatial reuse and mobility

The following notation is used:

- R: Instantaneous link data rate (bits/s)
- T: Time slot length (s), thus there are $1/T$ time slots per second and R/T bits/time slot.
- N: Number of nodes
- S: number of simultaneous sessions in the network
- N_{CDS}: Number of nodes in the CDS tree
- N_N: Rest of nodes (= $N - N_{CDS}$)
- R_C : Codec rate (bits/s), generated traffic rate including IP headers etc.
- A_N : number of assigned normal slots for each assigned normal node/second
- A_{CDS}: number of assigned CDS slots for a CDS node/second
- H_{CDS}: number of hops from source to destination through the CDS tree

The number of available time slots to a normal node is

$$A_N = Y/T/S \qquad \text{(case A)} \qquad (1)$$
$$A_N = Y/T/N_N \qquad \text{(case B).} \qquad (2)$$

Similary, the number of available nodes to a CDS node is

$$A_{CDS} = X/T/N_{CDS}. \qquad (3)$$

3 Capacity Requirements

The used codec rate, R_C, will give us a minimum needed capacity requirement on the transmitting nodes. In order to have a sufficient capacity on each of the nodes, $A_N > R_C/T/R$ timeslots/s for the assigned nodes, and $A_{CDS} > S * R_C/T/R$ for the CDS nodes must be fulfilled.

The capacity requirement for normal nodes can be written as $Y/S > R_C/R$ in case A and $Y/N_N > R_C/R$ in case B. This can be rewritten to

$$Y > S * R_C/R \qquad \text{(case A)} \qquad (4)$$

$$Y > N_N * R_C/R \qquad \text{(case B)} \qquad (5)$$

The capacity requirement for CDS nodes can be written as $X/N_{CDS} > S * R_C/R$ which can be rewritten as: $X > N_{CDS} * S * R_C/R$.

This is the minimum required capacity; however, if we only have capacity barely above this value, queuing delay may be significant as well, at least if we have multiple sessions being transmitted from a node. Queuing delay in the nodes will also depend to a large degree on the codec and how it is generating packets, if the voice codec generates packets of fixed size with a fixed interval, there will usually be no queues as long as we have sufficient capacity, but more advanced codec will try to minimize transmissions by sending only needed information based on the voice stream. This will decrease sent traffic, but average values may not necessary be good measures when determining required capacity. Nevertheless, as long as we have sufficient extra capacity, queuing delay will not be an issue. In the next section end-to-end delay is calculated assuming no packets in queue.

4 End-to-End Delay

The end-to-end delay of a voice session can be divided into three parts. First the sampling delay in the codec, ignoring computing delays we must still wait until we have sufficient time to fill a packet with all voice information generated between two time slots. With perfect sync between codec and MAC layer this would be the minimum delay incurred, but realistically this is not normally possible, so we also get some waiting time for our time slot as well which in average should be another half distance between time slots in addition to the previous value (or another full distance if we are unlucky). This second part can be seen as the delay of the first hop. (Notice, this assumes that the codec generates one packet per time slot. If smaller packets would be generated this delay would decrease, such a solution would come at a higher overhead cost though and would still give at least give the time difference between assigned time slots as a minimum delay of these two parts.)

Finally we also have the delay on the CDS tree. Assuming an equal delay on each CDS hop (both average $\bar{D}_{CDS}$ and maximum $\hat{D}_{CDS}$), the total delay on the CDS tree can be calculated as $\bar{D}_{CDS}$ or $\hat{D}_{CDS}$ multiplied with the number of CDS hops required.

This finally gives the average end-to-end delay as $D_{\text{avg}} = H_{CDS} * \bar{D}_{CDS} + 1/2/A_N + 1/A_N$, and the maximum delay as $D_{\text{max}} = H_{CDS} * \hat{D}_{CDS} + 1/A_N + 1/A_N$.

The average CDS node delay will be $1/2/A_{CDS}$ and the maximum CDS node delay $1/A_{CDS}$, resulting in the following average end-to-end-delay

$$D_{\text{avg}} = \begin{cases} H_{CDS} * T * N_{CDS}/2/X + 3 * T * S/2/Y, & \text{(case A)} \\ H_{CDS} * T * N_{CDS}/2/X + 3 * T * N_N/2/Y, & \text{(case B)} \end{cases} \tag{6}$$

And maximum delays as

$$D_{\text{max}} = \begin{cases} H_{CDS} * T * N_{CDS}/X + 2 * T * S/Y, & \text{(case A)} \\ H_{CDS} * T * N_{CDS}/X + 2 * T * N_N/Y, & \text{(case B)} \end{cases} \tag{7}$$

Both of these expressions are of the type $A/X + B/(1 - X)$ for which it can be shown a minimum occurs for $X = \sqrt{A}/(\sqrt{A} + \sqrt{B})$ and $Y = \sqrt{B}/(\sqrt{A} + \sqrt{B})$ at $A + B + 2 * \sqrt{AB}$.

The minimum average end-to-end delay D can be therefore be optimized in case A by choosing

$$X = \sqrt{H_{CDS} * N_{CDS}}/(\sqrt{H_{CDS} * N_{CDS}} + \sqrt{3 * S})$$

and

$$Y = \sqrt{3 * S}/(\sqrt{H_{CDS} * N_{CDS}} + \sqrt{3 * S}).$$

This will give a minimum delay of

$$D = T/2 * (H_{CDS} * N_{CDS} + 3 * S + 2 * \sqrt{H_{CDS} * N_{CDS} * 3 * S})$$

For case B we need to choose

$$X = \sqrt{H_{CDS} * N_{CDS}}/(\sqrt{H_{CDS} * N_{CDS}} + \sqrt{3 * N_N})$$

and

$$Y = \sqrt{3 * N_N}/(\sqrt{H_{CDS} * N_{CDS}} + \sqrt{3 * N_N}).$$

This will give a minimum average delay of

$$D = T/2 * (H_{CDS} * N_{CDS} + 3 * N_N + 2 * \sqrt{H_{CDS} * N_{CDS} * 3 * N_N}$$

The maximum delay is simply twice this value.

Time slot length will affect the delay directly, a doubling of time slot length will also double the delay. The data rates on the links will not affect the delay, however, unless we are capacity limited and thus will get queues. Realistically we will see some effect of the data rate though since we normally get some queues also at throughputs much lower than capacity, but the setting of time slot length will normally be a much more important tool for voice traffic.

5 Realistic values of parameters

In this section we will now study the capacity requirements and end-to-end delay for voice traffic when we set the parameter values. We will use the following range of parameter values

Parameters:

- R = [0,5 1] Mb/s (Values on the order of 1Mb/s seems reasonable for systems today.)
- CR = 5 kb/s (We set this arbitrarily to this value; it should give sufficient voice quality.)
- T = [2.5 5] ms (Below 2 ms may be difficult and there are no reason for time slots larger than 5ms,)
- N=[20 40 70] nodes (Small to a fairly large network.)
- N_{CDS}=1/3 of the nodes or $\frac{1}{2}$ of the nodes (Number of CDS nodes will vary depending on topology. For a fully connected network, the number of CDS nodes is much smaller but that is probably not a difficult scenario, at least not for case A)
- H_{CDS} = 4 (This gives 5 hops total, we would like to be able to handle at least 5 hops.)
- S = 3 (No more than 3 sessions seem reasonable in a single ad hoc network at the moment.)

5.1 Capacity requirements

Table 1: Capacity Requirements for different parameter settings

R [Mb/s]	N	N_{CDS}	C_A	C_B	$C_{CDS}(A)$	$C_{CDS}(B)$
0.5/1	20	6	0.08/0.04	0.24/0.12	0.33/0.16	0.46/0.23
0.5/1	40	13	0.10/0.05	0.49/0.24	0.55/0.27	0.88/0.44
0.5/1	70	23	0.13/0.06	0.84/0.42	0.94/0.47	1.6/0.8
0.5/1	20	10	0.09/0.05	0.22/0.11	0.44/0.22	0.56/0.23
0.5/1	40	20	0.12/0.06	0.43/0.22	0.8/0.4	1.12/0.56
0.5/1	70	35	0.15/0.07	0.75/0.37	1.3/0.65	1.9/0.9

In Table 1 we show how large part of the available capacity (assigned time slots) that will be used by the VoIP-session traffic depending on parameter setting and whether method A or B is used. C_A is the fraction of capacity used by assigned nodes in case A, C_B the fraction of capacity used in case B, and $C_{CDS}(A)$ and $C_{CDS}(B)$ the fraction of capacity used by the CDS nodes. The values for X and Y are chosen to minimize the delay which means in some cases (for example the 70-node networks) other choices would improve the capacity requirements, such values would increase delay though.

As can be seen, capacity is not so much of a problem for the assigned nodes (especially in case A) but it can be more of a problem for the CDS nodes, especially for large networks. With 500kb/s it is difficult to handle a 70 node network, and for the network with extra CDS nodes it is not even sufficient to give CDS nodes all available capacity. With the higher data rate and case A, all cases seem possible. For the higher data rate and case B we will still end up in capacity limitations at the 70-node networks but they are somewhat less severe.

5.2 End-to-end Delay

In Table 2, we show the resulting end-to-end delay for different parameter values. D_A is the average end-to-end delay in case A and D_B is the average end-to-end delay in case B.

Table 2: End-to-end delay for different parameter settings.

T [ms]	N	N_{CDS}	D_A [ms]	D_B [ms]
2.5/5	20	6	86/172	166/333
2.5/5	40	13	130/260	328/657
2.5/5	70	23	204/409	580/1160
2.5/5	20	10	109/217	174/348
2.5/5	40	20	178/357	348/696
2.5/5	70	35	275/550	609/1219

As can be seen the delay varies a lot depending on parameter values. For case A, using the short time slot length most delays would probably be acceptable in most cases. With longer time slot lengths and using case B delays may become problematic. How long delays that are tolerable depends a lot on the scenario of course. In some cases, a delay of 1 second or more is of little problem but it should be noted that in those cases we are also having capacity problems which means that delay will actually be even higher.

6 Spatial reuse, mobility and other complications

None of above calculations considers spatial reuse of time slots. By adding spatial reuse more slots could be given to each node which would decrease the average delay and increase the available capacity thus improving the voice service. In a perfect case, each assigned node and CDS node would get a proportionally higher number of slots, which would allow for an equivalent lower end-to-end delay or so many more simultaneous sessions. However, in practice number of our assumptions will be more difficult to achieve with the added spatial reuse:

- Time slots will be more difficult to spread at an equal distance when trying to handle spatial reuse,
- Giving all nodes a proportional number of slots compared to traffic load. Some nodes are more easy to give extra capacity, and central nodes may be interfered by many nodes and thus more difficult to assign an appropriate number of time slots.
- Handling mobility will be more difficult.

In the end, spatial reuse will result in an improvement (otherwise we wouldn't use it), but not as large as could be expected by just studying the increase in number of assigned nodes per time slot.

Mobility is another problem, as the nodes move the nodes in the CDS must change to keep up with the present topology. If the number of nodes in the CDS tree is constant, time slots could be given from a leaving node to the arriving new one quickly, but if the number of nodes increase this is not so simple (it may not be possible if we are using spatial reuse either). A quick way of relocating resources will be needed to avoid interruptions every time the CDS changes. One way to alleviate this problem can potentially be to choose the CDS for stability, i.e. to prefer static nodes and other nodes with more stable links. This will not resolve the problem though; rapid rescheduling will still be needed, just perhaps not so often.

7 Conclusions

Voice over multiple hops seems to be possible; both capacity requirements and delay requirements are achievable in most cases. However, it is clear that in order to make this work for larger networks there are some problems that needs to be considered. Capacity is a problem, three simultaneous sessions will strain the available capacity for larger networks. By only giving capacity to nodes that actually need it (case A) we will be doing better, but link capacities over 1Mb/s may be needed.

Time slot length has a direct effect on delay of voice traffic as long as it does not reduce the link data rate too much (something we have ignored here). The difference between 2.5ms and 5ms time slot length in table 2 is often the difference between acceptable and unacceptable for harsher requirements (150 ms). Keeping time slot length short should be an important feature of a VoIP waveform.

References

[1] M. Mastrogiovanni, A. Panconesi, and C. Petrioli, "Localized protocols for ad hoc clustering and backbone formation: A performance comparison," *IEEE Trans. on Parallell and Distributed Systems*, vol. 17, 2006.

[2] H. Liu, X. Jia, P.-J. Wan, X. Liu, and F. F. Yao, "A distributed and efficient coding scheme using 1-hop information in mobile ad hoc networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 18, no. 5, pp. 658–671, 2007.

[3] C. Adjih, P. Jacquet, and L. Viennot, "Computing connecting dominated sets with multipoint relays," *Ad Hoc & Sensor Wireless Networks*, pp. 9–43, March 2005.