Using Dual-Radio Nodes to Enable Quality of Service in a Clustered Wireless Mesh Network

Urban Bilstrup, Katrin Bilstrup, Bertil Svensson and Per-Arne Wiberg

Bilstrup U, Bilstrup K, Svensson B, Wiberg P. Using Dual-Radio Nodes to Enable Quality of Service in a Clustered Wireless Mesh Network. In: IEEE Conference on Emerging Technologies and Factory Automation, 2006. ETFA '06. Piscataway, N.J.: IEEE; 2006. p. 54-61.

DOI: http://dx.doi.org/10.1109/ETFA.2006.355184

Using Dual-Radio Nodes to Enable Quality of Service in a Clustered Wireless Mesh Network

Urban Bilstrup[†] Katrin Bilstrup[†] Bertil Svensson[†] Per-Arne Wiberg[†,‡]

[†]Centre for Research on Embedded Systems
Halmstad University, Box 823,
SE-301 18 Halmstad, Sweden

[‡]Free2move AB
Pilefeltsgatan 77
SE-302 50 Halmstad, Sweden

E-mail: {*Urban.Bilstrup*, *Katrin.Bilstrup, Bertil.Svensson}@ide.hh.se*, *Pelle@Free2move.se*

Abstract

In this paper some well established wireless technologies are merged into a new concept solution for a future industrial wireless mesh network. The suggested clustered wireless mesh network can handle probabilistic quality of service guarantees and is based on a dual-radio node architecture using synchronized frequency hopping spread spectrum Bluetooth radios. The proposed architecture gives a heuristic solution to the inter-cluster scheduling problem of gateway nodes in clustered architectures and breaks up the dependence between the local medium access schedules of adjacent clusters. The dual-radio feature also enables higher network connectivity, implying, for example, that a higher link redundancy can be achieved.

1. Introduction

Industry has recognized wireless communication as a solution to existing communication infrastructure problems [1, 2]. Two emerging technologies are wireless mesh networks [3] and wireless sensor networks [4]. Recent research results [5], standardization [6, 7] and the upcoming standard for WLAN mesh [8] are all very promising for wireless industrial communication. However, providing the necessary quality of service (QoS) guarantees for industrial communication is difficult with present technologies. Two clearly identified problems are; the fact that the wireless medium is error prone [9] and the fact that the channel assignment (broadcast scheduling) [10] is hard to achieve in a predictable way in wireless mesh networks.

The clustered topology, adopted from Ephremides *et al*. [11], divides the total set of nodes into sub sets (clusters). This topology provides a local and well defined architecture for channel assignment, medium access and spatial channel reuse. However, a clustered network topology does require a gateway node, Figure 1, which is able to participate in at least two clusters. The problem with this is that at each specific time instant, a device with a single transceiver is only able to transmit or receive in one cluster, i.e., simplex operation. As a result, the transceiver in the gateway node must be multiplexed (switched) between the clusters, e.g., time division multiplexing. This means that the local medium access scheduling inside each cluster becomes dependent on the neighboring clusters' medium access schedule, since a gateway node is a member of at least two clusters. This global time scheduling problem has an exponential complexity and there does not exist a solution with polynomial complexity [12]. It can also be concluded that a global sub-optimal solution is not an option for large networks [13, 14], since this implies that global network information would be needed. The only thing that seems to be practical and feasible in order to solve the inter-cluster scheduling problem of the gateway nodes is a local heuristic solution.

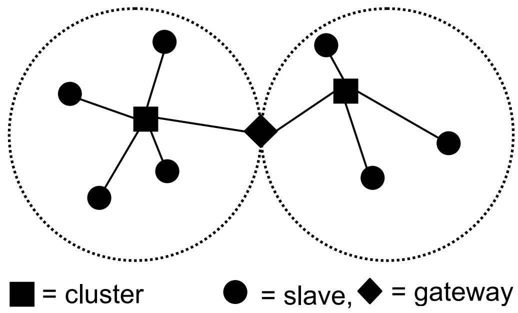

Figure 1. Cluster topology, with a gateway node shared by the neighbor clusters.

In this paper a clustered wireless mesh network based on a dual-radio gateway node is presented. The proposed dual-radio gateway node offers a heuristic solution to the gateway scheduling problem. This design enables probabilistic QoS guarantees to higher protocol layers, such as the networking and the application layer. Entities for QoS guarantees are provided by using the probabilistic framework proposed by Uhlemann *et al*. [15] and later on adjusted to Bluetooth by Bilstrup *et al*. [16]. In this framework the probability of successful delivery before deadline, $P_{success}$, and the response time, $T_{response}$, are given as QoS parameters.

The physical layer in the proposed wireless mesh network topology is based on the Bluetooth radio [17], e.g., using the same modulation technique as well as frequency hopping spread spectrum (FHSS). The master-slave polling scheme and the error detection scheme from the data link layer of Bluetooth have also been used.

2. Dual-radio architecture

In this section the dual-radio node using synchronized FHSS with external timing information is presented. Further, an inter-cluster scheduling method that avoids the near-far problem of dual-radio nodes is explained. Finally, the adopted clustering algorithm with the added FHSS-based control channel is also described.

2.1. Frequency Hopping Spread Spectrum

FHSS is an attractive technology for use in wireless mesh networks [18]. The frequency hopping provides resistance to fading, interference and hostile jamming [19]. The air interface is available from a standard Bluetooth transceiver [17], which uses a Gaussian frequency shift keying (GFSK) modulated digital radio signal, supporting FHSS. By using different hopping sequences FHSS creates multiple channels, enabling concurrent transmissions in neighboring clusters. If all neighboring clusters within the co-channel interference range were assigned orthogonal hopping sequences, no transmission collisions would occur. By using orthogonal sequences two users would never hop to the same frequency at the same time instant. However, the hopping sequence assignment problem is NP-hard [10]. The only practical solution is a local, heuristic method, where non-orthogonal hopping patterns are used, in our case by means of a random algorithm. Transmission collisions between concurrently operating neighbor clusters occur when the clusters utilize the same frequency channel in the same time slot. These collisions must be handled with appropriate fault tolerance, as described in Section 3.

2.2. Synchronized frequency hopping

We assume that all nodes in the network are time slot synchronized, e.g., by being equipped with a GPS receiver [20] or some other type of receiver that is able to receive broadcasted time marks. Using an external timing signal has proved to be a successful concept for time critical communication in self-organizing time division multiple access (STDMA) schemes. STDMA is used in, e.g., digital communication between aircrafts and between aircrafts and ground elements in the VHF Data Link 4 (VDL-4) system [21]. There is also an active maritime transponder system called Automatic Identification System (AIS) [22], which uses STDMA.

The multiplexing method proposed in this paper uses synchronized FHSS and an external timing signal to determine the hopping sequence for the FHSS scheme. The network is provided with a global synchronized beat that tells all nodes whether a time slot is odd or even and when it starts. The internal slot timing inside a cluster is done in the same manner as in a Bluetooth piconet [17] except that the external global time reference is used for slot synchronization of all nodes. The local time is generated by the global time reference to which an even discrete random offset is added, Figure 2. The addition of an even random offset is a precaution to ensure that the generated random hopping sequence is non-correlated with the hopping sequences of adjacent clusters.

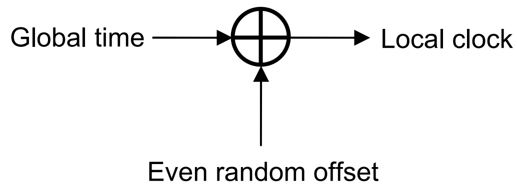

Figure 2. Local clock generation.

The local clock together with the address of the cluster head gives the local cluster's specific hopping sequence. The non-orthogonal hopping pattern is locally generated independently of the hopping patterns of adjacent clusters. All transceivers that are members of a cluster use the same identity (cluster head ID) and the same setting of random clock offset, Figure 3. When a node wants to become a member of a cluster it adjusts the random offset and identity settings of one of the transceiver to fit that specific cluster.

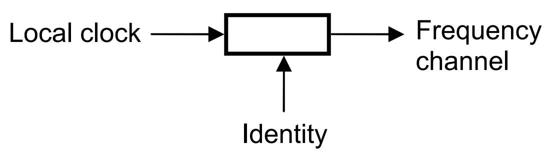

Figure 3. Frequency hopping pseudo-random generator.

2.3. Intra-cluster scheduling

The medium access inside each cluster is a master-slave polling scheme. This is used because of its simplicity and the fact that a dynamic scheduler with a cluster centralized acceptance test is applicable. From a scheduling perspective it is important that it is one single resource that is scheduled between multiple users. In this case the single resource is the broadcast channel, formed by the cluster's specific hopping sequence. The master-slave polling scheme is based on a time division duplex (TDD) scheme. The TDD scheme uses two time slots for each message exchange, one forward time slot and one reverse time slot. An information burst is transmitted from the cluster head during the forward time slot, whereas the subsequent time slot is reserved for the slave to transmit an information burst in the reverse direction. This master-slave polling mode of operation makes the channel sharing inside each cluster straightforward, since the cluster head provides a single point of operation. Each master-slave burst is triggered by the local schedule in the cluster head node according to some scheduling algorithm, e.g., earliest deadline first (EDF) [23].

2.4. Inter-cluster scheduling

Our heuristic solution to the gateway scheduling problem is based on the use of two separate transceivers in each node, Figure 4.

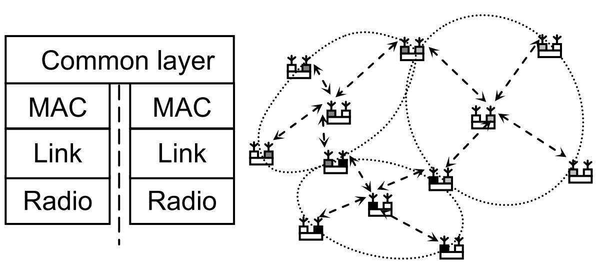

Figure 4. Each node is equipped with two transceivers.

This solution breaks up the dependence between the local medium access schedules of adjacent clusters. This will result in the gateway nodes not having to switch their presence between the two neighboring clusters. The protocol stacks of the two transceivers are merged together in a common layer above the link layer, as shown in Figure 4.

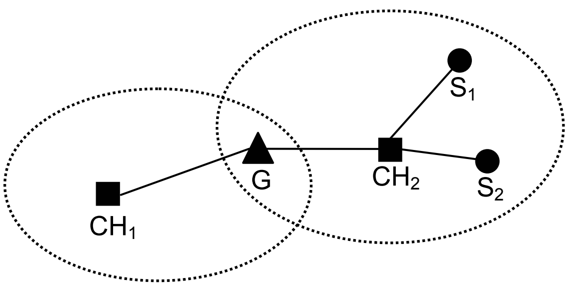

Figure 5. Two neighboring clusters sharing a gateway node equipped with two radios.

If the time slots between two adjacent clusters are not synchronized, the dual-radio solution will create node-local adjacent channel interference. If, on the other hand, they are slot synchronized, the two radio transceivers in the gateway node can be forced to receive or transmit in a synchronized way.

Consider the scenario when both transceivers in a gateway, denoted G #1 and G #2 are polled by the two cluster heads, CH_1 and CH_2, of two neighbor clusters, as seen Figure 5. For synchronized FHSS this happens at the same time instant, as can be seen in the transmission schedule in Figure 6. The synchronized FHSS thus eliminates the node's internal near-far problem since for each time slot G #1 and G #2 either both transmits or both receives. Consequently, the gateway node can be scheduled as any other node inside each individual cluster.

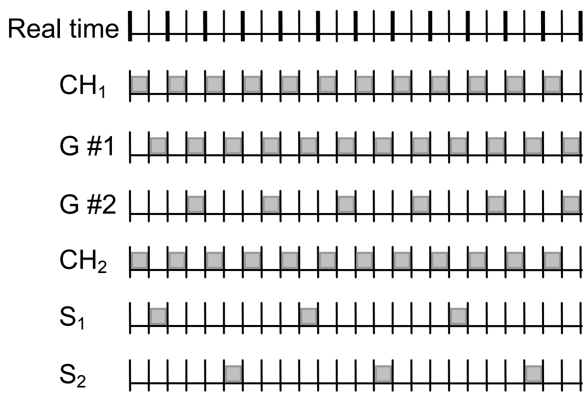

Figure 6. An inter-cluster schedule of a gateway node with two transceivers, applying synchronized transmissions.

2.5. Clustering

The formation of the network topology is done by a clustering process that forms feasible interconnected sub-networks out of the total set of nodes. All currently existing methods for clustering of nodes are based on heuristic algorithms and all future algorithms will also be based on this kind of sub-optimal algorithms. This statement is motivated by the fact that the optimal clustering problem is reducible to the computationally hard problem of finding the maximum independent set of nodes in a network graph [24, 25].

Two early suggested heuristic clustering algorithms are the lowest-ID algorithm [11] and the highest-connectivity algorithm [26]. In this paper the lowest-ID algorithm is adopted. In this algorithm, the identification number (ID) of the nodes is used to elect the head in a cluster. This algorithm chooses the cluster head on the spatial random distribution of the nodes, their IDs, and does not perform any kind of optimization. The algorithm works as follows:

- Each node is assigned an ID. Periodically, each node broadcasts the list of nodes (IDs) it can hear, including itself.

- A node only hearing nodes with higher IDs than itself will declare itself as a cluster head node.
- A node that can hear two (or more) cluster heads is a gateway between two clusters.
- A node that is neither a cluster head nor a gateway is an ordinary end node (slave).

The lowest-ID algorithm assumes a common broadcast channel; therefore a common control channel has to be added, described in the next sub-section.

2.6. Common control channel

The control channel is used during the clustering of nodes and is created by a common hop sequence. This sequence is actually a broadcast identity which is fed into the frequency hopping pseudo-random generator, Figure 3. A transceiver that wants to access part in the control channel sets its channel address to the broadcast identity and sets the random even clock offset to zero, as seen in Figure 2. All nodes use random access to broadcast hello messages on this channel. A hello message contains the node ID, the position, and the node state. The random access is based on a uniform distribution.

A node can be in different states: *not connected*, *fully connected gateway, cluster head,* or *end node*. If a node is in *not connected* state, it uses one of the transceivers to periodically broadcast hello messages and listen for other nodes' hello messages on the control channel. If the node is in *end node* state or in *cluster head* state, the non connected radio is used to broadcast hello messages and listen for other nodes' broadcasts. If the node is a *fully connected gateway* node it does not send any hello messages or listens for any other nodes' hello messages.

3. Quality of service guarantees

Since the synchronized FHSS with non-orthogonal hopping sequences is used, frequency collisions between adjacent clusters can occur. This must be taken into account when doing resource allocation. Following the method presented by Uhlemann *et al.* [15] it is possible to calculate the necessary resources to provide a certain probability of successful delivery. In this section we give a brief description of how this is done, for a more detailed description see Bilstrup *et al.* [16].

In Figure 7 the transportation of a single packet from master to slave (A.) and from slave to master (B.) is depicted. The master-slave polling scheme in combination with automatic repeat request (ARQ) and cyclic redundancy check (CRC) is used to indicate the correctness of received data and ask for retransmissions if a transmission is lost or corrupted.

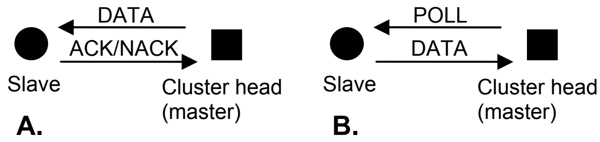

Figure 7. Packet exchange, point-to-point connection. A. Master to slave B. Slave to master.

3.1. Probability of correct delivery

Transmission failures occur when clusters, within the co-channel interference range, are using the same frequency. To provide guarantees for the correct data delivery, the probability of errors must be considered for each link. The collision rate is continuously monitored for each link inside a cluster. This statistics is used to estimate the appropriate amount of time redundancy that is needed for a specific link in order to raise the probability of correct delivery to an acceptable level.

The probability for collision in the two links involved in the transportation of a master to slave message has to be considered. First, DATA is forwarded from master to slave, and in reverse an ACK/NACK is transmitted from slave to master. In the other direction, slave to master transmission, a POLL packet is forwarded from the master to the slave and DATA is transmitted on the reverse link from the slave to the master. The probability of correct delivery before deadline, $P_{success}$, for all transmissions involved in the information transportation of a message can be represented as a permutation tree, Figure 8. The branches represent the individual transmissions' probabilities for collision and for no collision, respectively. The permutation tree in total, Figure 8, represents the probability of successful delivery (the black path in Figure 8), $P_{success}$, as well as the probability of unsuccessful delivery (the grey paths in Figure 8), P_{fail}.

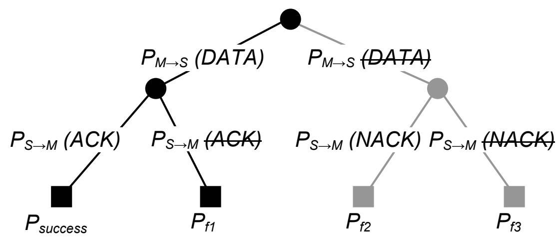

Figure 8. Permutation tree for probability of successful delivery, master to slave transmission.

3.2. Fault tolerance

In the permutation tree of Figure 8, there is only one successful path, i.e., the successful transmission time, T_t, is also the response time $T_{response}$. However, as can be seen in the permutation tree in Figure 9, when one redundant time slot is added several successful paths exist.

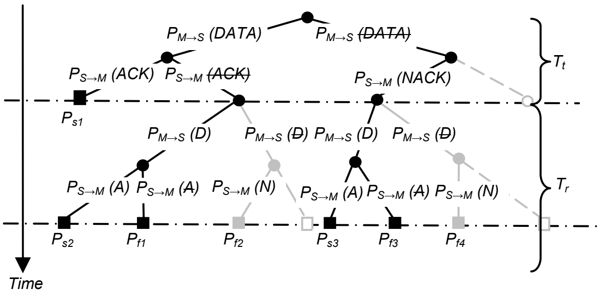

Figure 9. Permutation tree for a master to slave transmission, with one redundant frame added.

The probability of successful delivery $P_{success}$, is increased by adding an even number of redundant time slots, T_r, to the transmission time, T_t giving an increased response time:

$$T_{response} = T_t + 2T_r \, . \tag{1}$$

This gives a higher cumulative $P_{success}$. The probability of successful delivery for each individual successful path P_{si}, Figure 8, is added together according to:

$$P_{success} = \sum_{i=1}^{\forall \text{ successful paths}} P_{s_i} \, . \tag{2}$$

3.3. Cluster local resource allocation

When a new connection is set up the resources inside a cluster are negotiated with the next higher layer, i.e., the network layer. A certain connection requires certain QoS guarantees. These requirements are locally represented as $T_{response}$ and $P_{success}$. The request is given to the cluster head node. This starts with calculating the necessary resources (i.e. time slots) that are needed to reach the QoS for the connection request by using $T_{response}$ and $P_{success}$. After this the cluster head performs a feasibility test to see whether there are enough resources in the whole cluster to admit this new connection or not. If not the cluster head will renegotiate the QoS with the requesting slave. If there are enough resources the cluster head will schedule the new connection among the other communication tasks using some scheduling algorithm, e.g., earliest due date [27] or earliest deadline first [23].

The $T_{response}$ for a certain $P_{success}$ is represented by the depth of the tree, Figure 8. If the resulting $P_{success}$ does not fulfil the needed QoS guarantee on successful delivery, the $T_{response}$ is increased, i.e., a new layer in the permutation tree is added. New successful paths are calculated and $P_{success}$ is once again compared to the successful delivery guarantee asked for and so forth until the required guaranteed QoS level is reached or a maximum level of $T_{response}$ is reached. If the maximum level of $T_{response}$ is reached, i.e., not enough resources are available, the QoS must be renegotiated with higher protocol layers.

4. Simulation

Some initial simulations have been conducted. The first investigation reveals which impact the co-interference has on the probability of transmission collision. The second evaluates how the addition of redundant time slots for fault tolerance can be used to increase the probability of successful delivery.

4.1. Simulation model

A discrete event simulator has been developed and used for the performance evaluation. The simulation assumes a 1000 x 1000 meter square area. The (X_j, Y_j) position for node j is drawn from uniform distributions [0, X_{max}] and [0, Y_{max}]. The cluster formation is performed with the lowest ID clustering algorithm [14]. The signal-to-interference ratio (SIR) [28], is the ratio between the received signal energy P_r and the sum of the energy of all interfering signals, I_i, plus the thermal noise N_0, where index i is defined as *the set of all nodes that transmit concurrently utilizing the same channel*, according to

$$SIR = \frac{P_r}{N_0 + \sum_{i \in transnodes} I_i} \, . \tag{3}$$

The interference level at a certain point in time and space is a direct consequence of the power used by all concurrently ongoing transmissions and the actual positions of the transmitting nodes. Furthermore, the necessary transmission power is a consequence of the distance (signal path loss) between the transmitting and the receiving nodes. The average large-scale path loss, L_p, for an arbitrary transmitter-receiver separation is a function of the distance r and the path-loss rate exponent n [28],

$$L_p(r,n) = L_f(r_0) + 10n\log_{10}\left(\frac{r}{r_0}\right) \tag{4}$$

where r_0 is a reference distance for the far field. The path loss exponent, n, indicates the rate at which the path loss increases with the distance, r. It can vary between 2 and 6, where $n < 2$ is free space propagation, $2 \le n \le 3$ is obstruction in factories and $4 < n \le 6$ is obstruction in home/office environments. In [29] it is shown that the path loss in factory buildings is lower than for the home environment. The r_0 distance should be greater than the near field of the antenna (Fraunhofer distance), approximately equal to one meter in our case. The path loss up to that reference distance, r_0, is either determined from a measurement close to the transmitter antenna or by the free space propagation loss equation according to:

$$L_f(r) = \frac{G_t G_r \lambda^2}{4\pi^2 r^2} \tag{5}$$

where G_t and G_r [28] are the antenna gains at the transmitter and receiver, respectively. In a wireless mesh network the antenna gain is preferably unity. The wavelength, λ, is given by the carrier frequency in use. The received power, P_r, [28] can be calculated according to:

$$P_r = 10\log_{10}(P_t) + L_p(r,n)\ . \tag{6}$$

The interference level, I, has an impact on the necessary transmission power since a certain SIR level is necessary for the receiving node in order to correctly receive the transmission. The interference, I_i, [28] from a concurrently transmitting node, i, using the same frequency, is calculated as:

$$I_i = 10\log_{10}(P_i) + L_i(r,n)\ . \tag{7}$$

The SIR is assumed to be above 15 dB at the receiver, in order for a transmission not to be counted as a collision. The physical layer uses FHSS with a hop rate of 1600 hops per second and all clusters are using master-slave polling. In this simulation each master polls the other member nodes of the cluster in a round robin fashion; two time slots are used for each node in each round (forward/reverse message exchange). All clusters are further assumed to be operating continuously.

4.2. Collision probability

In Figure 10 the number of clusters versus the probability of transmission collision for different path loss exponents is displayed. The simulation only considers transmission collisions between clusters within the co-channel interference range and thus no consideration is taken about failures that are caused by bit errors.

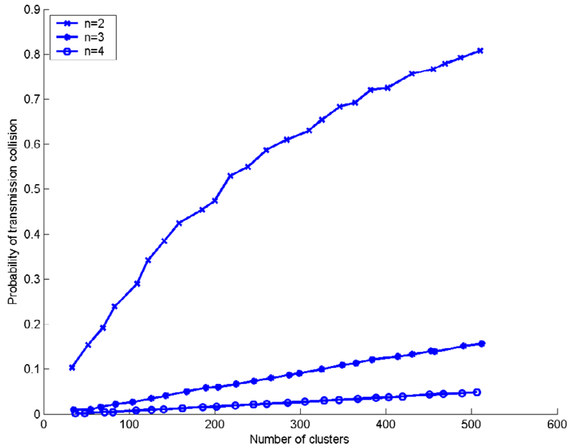

Figure 10. Probability of transmission collision versus number of clusters.

An important observation is that the probability of transmission collisions is increasing differently for different values of the path loss exponent. For high values of n the probability of transmission collisions is increasing at a much lower rate. It is obvious that a high path loss exponent is profitable for spatial channel reuse in a wireless mesh network.

4.3. Fault tolerance

The graph in Figure 11 displays the probability of failure P_{fail} versus the probability of transmission collision, for different numbers of redundant time slots. The case with only two allocated time slots (dotted curve) is when $T_t = RT$, i.e., no redundant time slots are added. The other curves show P_{fail} when two, four and eight redundant time slots are added to the RT.

The graph in Figure 11 indicates that it is possible to achieve low probability of missed deadlines, P_{fail}, even if the probability for a transmission collision is rather high. It can also be concluded that if the probability of collision is above 0.1, a large number of redundant time slots must be added to achieve a low probability of missed deadlines. In the graph in Figure 10 it can be seen that a probability of transmission collision above 0.1 is the situation for nodes with a path loss exponent of 2.

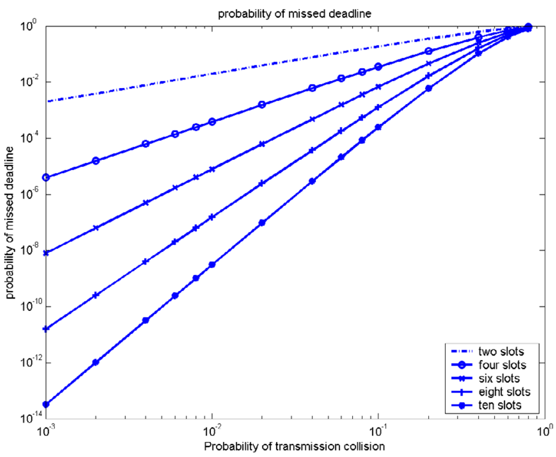

Figure 11. Probability of missed deadline versus probability of transmission collision.

However, one can see from Figure 10, that when the path loss exponent is greater than 3, the probability of transmission collisions is below 0.1 for up to 300 nodes in the 1000 x 1000 m square area and for up to at least 500 nodes if the path loss exponent is larger than 4. It should also be pointed out that it is a very pessimistic view since in this simulation all clusters are assumed to be fully loaded. The actual probability of transmission collisions will be lower if real traffic were injected into the simulation, i.e., all clusters would not be fully loaded all the time.

5. Conclusion

A clustered network topology requires gateway nodes for interconnecting neighbor clusters. In order to provide inter-cluster connectivity, a gateway node must concurrently participate in at least two clusters. To fully utilize the benefits provided by clustered networks, we suggest using a dual-radio node architecture, in which the two sub-radio units are participating in separate neighboring clusters. The dual-radio node topology prevents the local cluster medium access and traffic schedule from dictating the behavior of neighboring clusters and hence breaks up the inter-cluster dependence. The use of a local heuristic solution to the inter-gateway scheduling problem minimizes the demand for control information exchange between clusters. The dual-radio feature also enables higher network connectivity, since an sub-radio currently not used in a cluster can broadcast hello messages. This may lead to, e.g. that a higher link redundancy can be achieved.

Non-orthogonal hopping patterns introduce occasional frequency collisions resulting in lost transmissions. We propose a QoS mechanism to handle these collisions, adding appropriate amount of fault tolerance for each connection.

The simulation results reveals that when the system is running under pessimistic circumstances (the path loss exponent is equal to 2 and all clusters are fully loaded with traffic), the dual-radio node architecture works well for small-sized networks (below a hundred nodes). However, when the path loss exponent is larger than 3, which is a more realistic assumption, the dual-radio node architecture with random frequency hopping assignment and the fault tolerance mechanism works well even in middle-sized mesh networks (up to a couple of hundred nodes). Furthermore, it should be noted that if a more realistic traffic model is used, instead of the fully loaded clusters, an even better result would be achieved. It is likely that we could scale up to a large sized network with several thousands of nodes.

References

[1] T. S. Rappaport, “Indoor Radio Communications for Factories in the Future,” *IEEE Commun. Mag.*, vol. 27. issue 5, May 1989, pp. 15-24.

[2] P-A. Wiberg and U. Bilstrup, “Wireless technology in industry – applications and user scenarios,” in *Proceedings of the 8th IEEE Conference on Emerging Technologies and Factory Automation*, Antibes-Juan les Pins, France, October 2001, pp. 123-133.

[3] K. Koumpis, L. Hanna, M. Andersson, and M. Johansson, “A review and roadmap of wireless industrial control,” available at: http://wireless.industrial-networking.com/articles/articledisplay.asp?id=913 (February 2006).

[4] R. Zurawski ed., *The Industrial Information Handbook*, CRC press, U.S., 2005.

[5] A. Willig, K. Matheus and A. Wolisz, “Wireless technology in industrial networks,” in *Proceedings of the IEEE*, Vol. 93, No. 6, June 2005 pp. 1130-1151.

[6] IEEE Computer Society LAN MAN Standards Committee. *IEEE Std 802.15.4-2003 Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs)*. The Institute of Electrical and Electronics Engineers, New York, NY, May 2003.

[7] ZigBee Alliance, *ZigBee standard v 1.0*, 2005.

[8] G. Held, *Wireless Mesh Networks*, Auerbach Publications, New York, U.S., 2005.

[9] U. Bilstrup and P-A Wiberg, “Bluetooth in industrial environment,” in *Proceedings of the 11th IEEE International Workshop on Factory Communication Systems*, Porto, Portugal, September 2000, pp. 239-246.

[10] R. Ramanathan “A unified framework and algorithm for (T/F/C)DMA channel assignment in wireless networks,” in *Proceeding of INFOCOMM 97*, Kobe Japan April 1997, pp 900-907.

[11] A. Ephremides, J. E. Wieselthier, and D. J. Baler, ”A design concept for reliable radio network with distributed control,” in *Proceedings of the IEEE 75(1)*, 1987, pp. 56-73.

[12] M. Kazantzidis and M. Gerla. “On the impact of inter-piconet scheduling in Bluetooth scatternets,” in *Proceedings of The 3rd International Conference on Internet Computing IC 2002*, Las Vegas, U.S., June 24-27, 2002 pp. 37-43.

[13] U. Bilstrup, K. Sjöberg, B. Svensson and P-A.Wiberg “Capacity limitations in wireless sensor networks,” in *Proceedings of ETFA 2003*, *9th IEEE International Conference on Emerging Technologies and Factory Automation*, Lisbon, Portugal, 16-19 September, 2003, vol. 1, pp. 529-536.

[14] U. Bilstrup, B. Svensson and P-A. Wiberg “Knowledge horizon – dynamic limitations in a wireless ad hoc network,” in *Proceedings of 1st Swedish National Computer Networking Workshop (SNCNW2003)*, Stockholm, 8-10 September 2003, pp. 83-86.

[15] E. Uhlemann, T. M. Aulin, L. K. Rasmussen and P-A. Wiberg, “Deadline dependent coding – a framework for wireless real-time communication,” in *Proceedings International Conference on Real-Time Computing Systems and Applications*, Cheju Island, South Korea, December 2000, pp. 135-142.

[16] U. Bilstrup , K. Sjöberg B. Svensson, and P-A. Wiberg, “A fault tolerance test enabling QoS in a Bluetooth piconet,” in *Proceedings of the 2004 International Workshop on Real-time Networks*, RTN 2004, Catania, Italy, June 2004, pp. 47-51.

[17] Bluetooth SIG. *Specification of the Bluetooth system – Core v. 1.1*, February, 2001.

[18] A. Epheremides and J. E. Wiselthier, “A design concept for reliable mobile radio networks with frequency hopping signaling,” in *Proceedings of the IEEE*, January 1987, pp. 56-73.

[19] R. C. Dixon. *Spread Spectrum Systems with Commercial Applications*, John Wiley & Sons, U.S., 1994.

[20] E. D. Kaplan. *Understanding GPS: Principles and Applications*. Artech House, 1996.

[21] NEAN Update Program (NUP). *VDL MODE 4 Implementation and Operating Principles*, The North European automatic dependent surveillance- broadcast network update program, November, 2002, available from www.nup.nu (060215).

[22] International Association of Lighthouse Authorities, “Technical characteristics for a universal ship borne automatic identification system using time division multiple access in the maritime mobile band,” *International Association of Lighthouse Authorities IALA*, 2001.

[23] C. L. Liu and J. W. Layland, “Scheduling algorithms for multiprogramming in a hard real-time environment”, *Journal of ACMI,* vol. 20:1, 1973, pp. 46-61.

[24] M.R. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP – Completeness*, Freeman, San Francisco, CA, US, 1979.

[25] S. Basagni, *On the Broadcast and Clustering Problems in Peer-To-Peer Networks*. Ph.D. dissertation, Univ. degli Studi di Milano, Milano, Italy, May 1998.

[26] M. Gerla and C. R. Lin, “Multimedia transport in multihop dynamic packet radio networks,” in *Proceedings of International Conference Network protocols*, Tokyo, Japan, 7-10 November 1995, pp. 209-216.

[27] D. Ferrari and D. C. Verma, “A scheme for real-time channel establishment in wide-area networks,” *IEEE Journal on Selected Areas in Communications*, no. 8, 1990, pp. 368-379.

[28] T. S. Rappaport. *Wireless Communication*, Prentice Hall, New Jersey, U.S., 1996.

[29] T. S. Rappaport, “Indoor radio communications for factories of the future,” *IEEE Communications Magazine*, vol. 27, no 5, May 1989, pp. 15-24.