Context-Aware QoS Control for Wireless Mesh Networks of UAVs

Patrick-Benjamin Bök and York Tüchelmann
Department of Electrical Engineering and Information Sciences
Research Group Integrated Information Systems
Ruhr-University Bochum
Bochum, Germany
Email: patrick@research.boek.de, york.tuechelmann@iis.rub.de

***Abstract*—Unmanned Aerial Vehicles (UAVs) can be used in a wide application range. For example, UAVs are utilized to observe critical areas in disaster situations without jeopardizing individuals or to extend the transmission range of communication networks. Connecting a set of UAVs with each other within a Wireless Mesh Network helps to increase the coverage of the observable area. Due to their limited performance and energy resources and, thereby, restricted communication capabilities, a tailored QoS management scheme has to be used to optimize occurring data flows in these networks. We present a QoS control scheme that works on the basis of process-patterns, each describing the context-dependent behavior of UAVs according to the execution order of services for different situations. Furthermore, the logical communication path is optimized within the mesh network based on each node's areal position using a dynamic hierarchical communication structure. Thereby, performance and fairness within a network of UAVs can be increased demand-actuated.**

***Index Terms*—Hierarchical QoS Control; Process-Pattern; QoS Management; Unmanned Aerial Vehicles; Wireless Mesh Networks.**

I. Introduction

Unmanned Aerial Vehicles (UAVs) are an emerging technology that allows fast response in disaster situations without the use of individuals in critical areas [1], [2]. UAVs may be equipped with cameras, different sensors or communication modules. A set of UAVs may be connected with each other and a base station (BS) using wireless communication technologies like IEEE 802.11, WiMAX or LTE to a Wireless (Mesh) Network of UAVs (WNUAVs). Thus, UAVs allow to observe wide areas [3] affected by a disaster very fast, which is critical in these scenarios, and without jeopardizing a huge amount of personnel.

The problem of QoS management and congestion control known from general wireless networks is also present for the setting of UAVs, especially if these form a Wireless Mesh Network (WMN) used to extend the areal covering [4]. Also the mobility of UAVs is a problem that QoS management schemes have to face up [5]. There has not been a lot of effort to improve QoS management and congestion control for this special kind of network. But a lot of research exists for similar wireless environments like Wireless Sensor Networks (WSN) or Wireless Mesh Networks (WMN). A brief review of these researches is given in Section II. As documented in the related work, improving the performance of Wireless Mesh Networks on ISO/OSI Layers 1 and 2, especially for UAVs, is one alternative. Nevertheless, a tailored context-aware performance management can only be performed on upper layers. The use case of UAVs is different to other applications of Wireless Mesh Networks, due to their dynamics and their field of application. Therefore, different aspects like building and controlling the communication structure of the UAVs or to control the UAVs dependent of their current, situation dependent tasks, have to be included within a tailored QoS control scheme.

For UAVs, their controllable behavior can be used to drive the QoS management. Therefore, process-patterns that represent an ordered composition of tasks (e.g. the behavior or processes that are executed in disaster situations or at mega events) are introduced. Process-patterns are in some parts similar to workflows that we introduced in [6] to build a business process driven QoS provisioning scheme. Nevertheless, workflows and that scheme suffer from the problem that non-deterministic user behavior has to be considered. This problem is negligible because in most cases predefined patterns occur if using UAVs. Consequently, process-patterns are the deterministic subset of the non-deterministic workflows and allow to use an optimized structure.

In this paper, we present a context-aware QoS control scheme for Wireless Mesh Networks of UAVs based on process-patterns. Beside QoS control, our scheme includes an optimized hierarchical communication structure that reduces QoS and also command control overhead, while allowing a group based steering of UAVs. The structure of process-patterns is introduced in Section III. The setup of the dynamic hierarchical communication structure of the WNUAVs is presented in Section IV, before the proposed context-aware QoS control scheme is introduced and the occurring benefits for WNUAVs are discussed in Section V. Finally, we conclude our contribution in Section VI.

II. Related Work

Pattara-Atikom et al. [7] improve the distributed coordination function of IEEE 802.11 and raise the fairness in wireless networks. Therefore, algorithms for distributed fair queuing are presented. Also Luo et al. [8] present an algorithm for distributed fair scheduling in multi-hop ad hoc networks and show that their algorithm achieves high resilience against erroneous scheduling. However, Kong et al. [9] analyze the impact of selfish behavior on the performance of packet scheduling in wireless mesh networks and propose a game theoretic approach to optimize packet scheduling. Lee et al. [10] provide a practical solution for the configuration of the parameters in IEEE 802.11e, based on upper layer requirements, and present a cross-layer approach that improves the quality of multimedia traffic. Zhai et al. [11] and also Luo et al. [12] present different multi-hop packet-scheduling algorithms, each trying to achieve maximum throughput for traffic-flows in shared channel environments. Cheng and Zhuang [13], [14], [15] present a node-clustering algorithm with tax-based power allocation for wireless mesh networks. Their algorithm raises the trade-off between throughput and fairness in WMNs, which support QoS provisioning. In addition, Leu and Hsieh [16] also improve the performance of WMNs using a node-clustering algorithm. Especially in wireless sensor networks, their approach allows the absence of a central sink, which improves the autonomy of the wireless sensor network.

Marwaha et al. [17] present a distributed end-to-end allocation method, which allows to allocate time-slots for real-time traffic in WMNs. Time-slots will be reserved on a certain path. All neighbors on the path will be included within the reservation process. Thereby, interference, caused by reachable neighbors, can be pro-actively reduced. Finally, Liu et al. [18] introduce an enhanced per-flow admission control and QoS provisioning scheme for WMNs. The assignment of different channel access parameters for each flow improves the performance and fairness in WMNs.

All of these approaches try to achieve global fairness between nodes. Thereby, a better overall performance is obviously achieved. Nevertheless, the specific, most of the time situation dependent, requirements based on the processes performed on the wireless nodes - in this case the UAVs - are neglected. Using these approaches may lead to global fairness, but not to a dynamic and incident driven QoS control and, thus, they are not based on real QoS requirements of each incident or process as it is required, for example in disaster situations.

III. Process Patterns

The proposed context-aware QoS scheme is primarily based on process-patterns. Process-patterns represent certain processes that describe how to fulfill a superior task that is subdivided into small, sequential or parallel tasks. Each task represents a service that is used to achieve a certain objective in the context of UAVs. Pre-defined process-patterns are the only required input for the scheme to work autonomously. A process-pattern consists of one or more tasks. Each task holds information about the associated service. Also a set of target information about the generated packets in each task is included. Additionally, each task holds a QoS priority value that is used to globally weight tasks among each other. An example of a simplified process-pattern for a UAV fielding scenario is illustrated in Fig. 1.

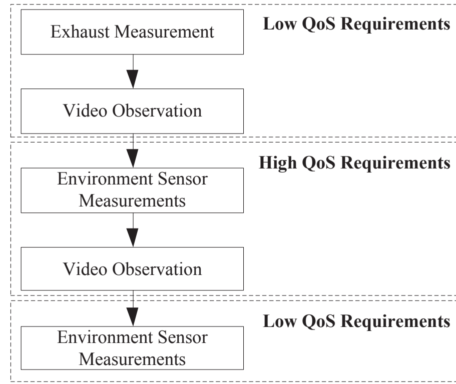

Fig. 1. Process-Pattern (Example)

Consider a disaster situation (e.g. a fire within a factory) in which UAVs are stationed above the affected area. As a first task, maybe the exhausts will be measured and the results will be sent to a central controller. Afterwards, a thermography video observation will start in order to recognize trouble spots and also individuals that maybe within the disaster area. This second task runs until an individual is detected. It has been defined that both tasks only have low QoS requirements. After a person has been recognized, environment sensor measurements (e.g. for temperature or coordinates) will be performed. If an individual has been detected in this disaster situation, the data generated in the following has to be sent to a central controller having high QoS requirements. Hence, also the fourth task, which again includes a thermography video observation, now runs having a higher priority than before because visual information that help to find a person are of great importance. After this task has finished, the next task (environment sensor measurements) will be performed, again having low QoS requirements.

The advantage of using process-patterns is that they allow an execution context based management of the desired level of QoS. A service having the same system process, the same destination, destination port etc., having different demands on QoS over time, can be identified by the serices that ran before or even run in parallel. This does not work if service, destination etc. based approaches are used. So, if process-patterns of all desired processes for UAVs in disaster situations are recorded, an autonomous QoS management in Wireless Networks of UAVs becomes possible. Thus, intelligent traffic shaping becomes viable and the limited resources, especially of nodes that have to forward network traffic within a Wireless Mesh Network, can be saved.

IV. UAV Communication Structure

The communication paths between UAVs and their central controller in a Wireless Mesh Network have to be optimized to achieve benefits in performance and energy-consumption. Therefore, we use a dynamic hierarchical group communication structure on the basis of each UAV's areal position. This optimization is motivated by our results from research on integrated network control architectures. In [19], we have shown that the performance of integrated network control architectures can be significantly improved with our proposed group communication structure, built upon the physical structure of the network.

Due to the low capabilities and constraints regarding energy consumption of UAVs, these existing flows as well as the additional QoS control flows between a central controller and the UAVs may affect the performance and the resilience of the WNUAV. Instead of using a flat WMN structure, which is shown in the upper part of Fig. 2, we use a hierarchical group communication structure as shown in the lower part of Fig. 2. Beside flow optimization, also the wireless link quality of the UAVs will be optimized. UAVs in the outer ring, which usually have poorer link quality, will gain a much better link quality because the next hop is closer to them. If using a dynamic transmission power control scheme, also UAVs in the inner ring will gain a much better link quality because interference may be reduced.

The dynamic hierarchical group communication structure will be constructed dynamically dependent of the areal position, computation capacity, energy consumption of the UAVs and the wireless link quality towards the central controller, instead of simply building a wireless mesh network. Therefore, groups of geographically close UAVs are built. The autonomously built structure allows to control the UAVs by a central Master, while improving the overall performance and the coverage of the WNUAVs. Each group consists of a group leader (SuperNode) that has the best connection quality to the Base Station (BS), among other things, and that is logically connected to a central Master controlling a sub-swarm of UAVs. The SuperNode is responsible for the communication between itself and the other UAVs in its group with the central Master. Using this concept, the central Master builds a logical hierarchy upon the WMN hierarchy. Tasks that affect a group of nodes do not have to be committed to each of the nodes by the central Master because this can be delegated to the SuperNodes. Therefore, bottleneck situations that may occur at the BS can be reduced or even avoided. If a SuperNode (UAV) changes its position significantly, the communication structure will be updated immediately and one of the other group members will be chosen as the group leader. The costs for updating the logical communication in such a case are negligible because only the assignment of the subordinated nodes has to be changed and announced. This is also the case if a subordinated node changes its position significantly because only the group assignment has to be changed and announced.

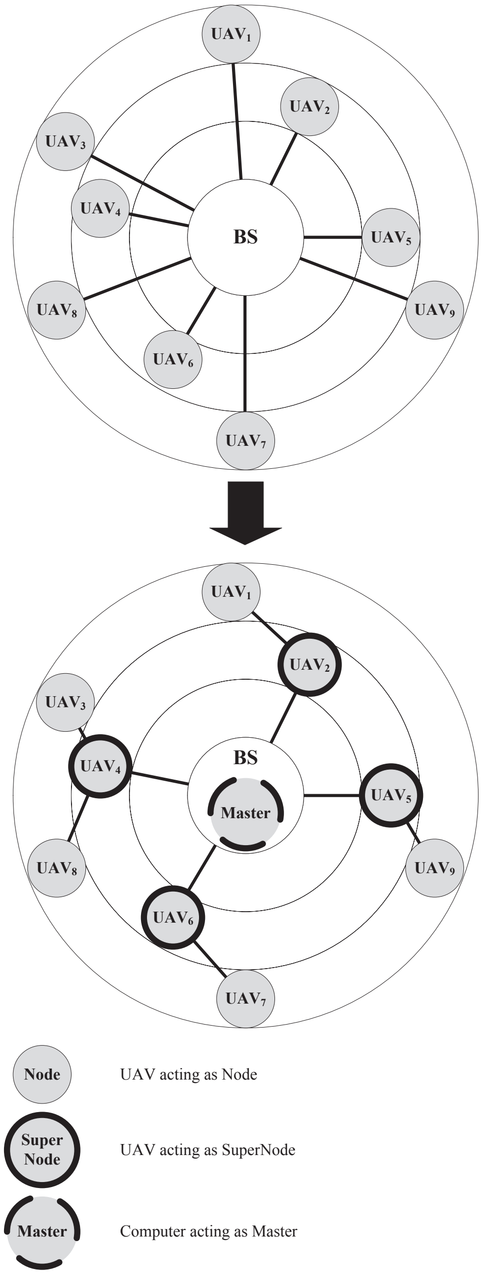

Fig. 2. Communication Structure

V. Process-Pattern Aware QoS Control Scheme

The main component of the proposed scheme is the autonomous QoS management, presented in Section V-A. Instead of using flow charts, the scheme and its behavior are presented using an exemplary sequence description in Section V-B. The benefits of our scheme are discussed in Section V-C.

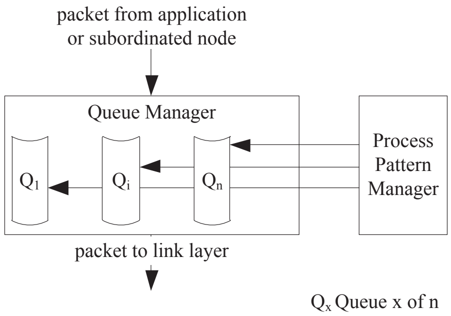

Fig. 3. Overview: QoS Management

A. Autonomous QoS Management

The autonomous QoS Management, which allows for process-pattern driven QoS provisioning in hierarchical WNUAVs, consists of two components (see Fig. 3): *Queue Manager* and *Process-Pattern Manager*. It can be run in two different modes: *Standard Mode* or *SuperNode Mode*. In the Standard mode, which runs on UAVs that are not SuperNodes, the Queue Manager has to queue its own outgoing traffic. In the SuperNode mode, which runs on UAVs that are utilized as SuperNodes, the *Queue Manager* also has to handle the forwarding traffic of its subordinated nodes. The queuing is performed according to the information provided by the *Process-Pattern Manager*. This component manages all existing process-patterns and identifies the current active service task of a process-pattern. The number of weighted queues is dynamically changed according to the number of different active flows.

Every outgoing packet is fetched by the Queue Manager, which calls the Process-Pattern Manager to determine the process-pattern that belongs to the flow and identifies its current active task. Afterwards the Queue Manager modifies the Type-of-Service (ToS) field in the IP-Header of the fetched packet and puts it into a certain queue. Following packets of a flow are queued much faster using system internal flow labels. Thereby, the identification of the process-pattern is improved significantly. Uncoupled from that, the Queue Manager also handles the queues using an appropriate scheduling algorithm and sends out the packets.

In the SuperNode mode, the behavior is similar to the behavior in Standard mode, but with the difference that it allows to handle forwarding traffic. Packets that belong to this kind of traffic are also fetched, as described before, but only the value of their Type-of-Service (ToS) field is analyzed and the packets are queued accordant to these values. Thereby, outgoing and forwarding traffic can be shaped according to process-patterns.

B. Sequence Description of Group Based QoS Control

A *Command Controller* represents another component in our scheme. It controls process-patterns as well as the group communication features. Furthermore, it is responsible for traffic forwarding on SuperNodes and gains the information required by the Master to build up the communication hierarchy. It is important to note that we differentiate between the process-patterns triggered by the Command Controller and the context-aware process-patterns that drive the behavior of the Process-Pattern Manager and the Queue Manager in the autonomous QoS management. Both do not influence each other. Thus, the QoS provisioning works autonomously from the running services on UAVs.

In the following, the pattern x_0 (see Fig. 4) is used to demonstrate the scheme's behavior. The pattern consists of four tasks that are partially using the same service having the same target information. This leads to the problem how to decide, which of a service's generated traffic has low and which of the traffic has high QoS requirements.

After the hierarchical group communication structure has been built by the Master, the Command Controller is able to send group affecting tasks towards SuperNodes, as illustrated in Fig. 5. The Master sends a command to start the pattern x_0 to UAV_1 (SuperNode) at t_0. The Command Controller of UAV_1 will then start the according pattern and forward the command to the members of the group, UAV_2 and UAV_3, whose Command Controller will also start pattern x_0. UAV_1 to UAV_3 will then start task T_1 of x_0, each. Additionally, UAV_1 forwards the data generated by UAV_2 and UAV_3. From t_0 to $t_0 + \delta_1$ task T_1 generates data that is sent to the Master, as illustrated in Fig. 6. This traffic of service X in task T_1 has low QoS requirements. At $t_0+\delta_1$ the Master triggers the UAVs to perform the next task. Each Command Controller will then try to start the next task and its according service Y (see Fig. 7). Consider a case in which not every subordinated UAV of a SuperNode has already finished its task when it is triggered. As illustrated in Fig. 8, from $t_0 + \delta_1$ to $t_0 + \delta_2$ UAV_3 still sends data of task T_1 in parallel to data of task T_2, while all other UAVs already stopped sending data of task T_1.
It is no problem to differentiate between the traffic of tasks T_1

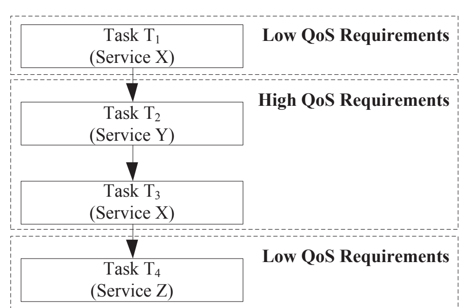

Fig. 4. Exemplary Process-Pattern x_0

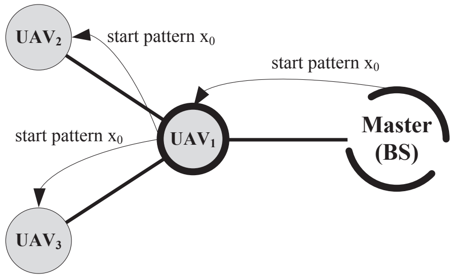

Fig. 5. Behavior at t_0

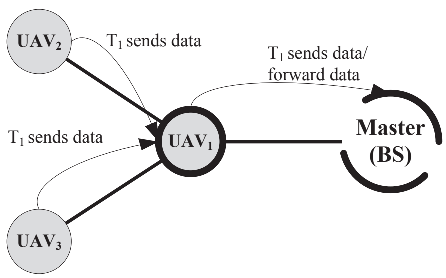

Fig. 6. Behavior from t_0 to $t_0 + \delta_1$

and T_2 when these are sending in parallel. At $t_0{+}\delta_2$ the Master triggers the UAVs to perform the next task and each UAV switches towards task T_3. Regarding the process-pattern (see Fig. 4), task T_3 again runs service X that generates network traffic from $t_0 + \delta_2$ to $t_0 + \delta_3$ (see Fig. 9), but now the service has high QoS requirements. However, it is impossible for common approaches to differentiate between parallel traffic of tasks T_1 and T_3 because both tasks use the same service with the same target information, but different QoS requirements.

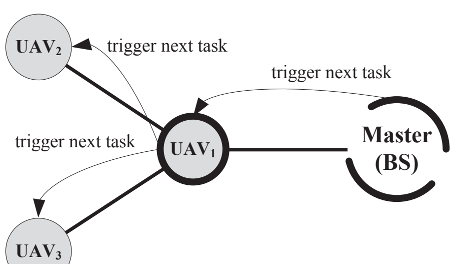

Fig. 7. Behavior at $t_0 + \delta_1$

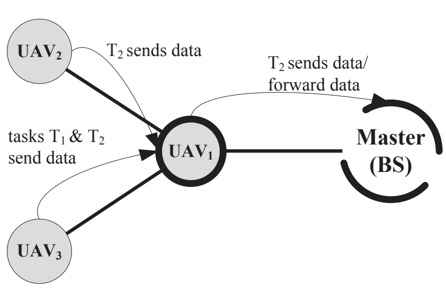

Fig. 8. Behavior from $t_0 + \delta_1$ to $t_0 + \delta_2$

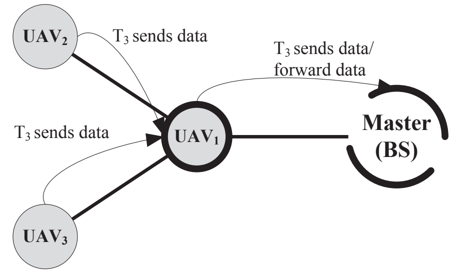

Fig. 9. Behavior from $t_0 + \delta_2$ to $t_0 + \delta_3$

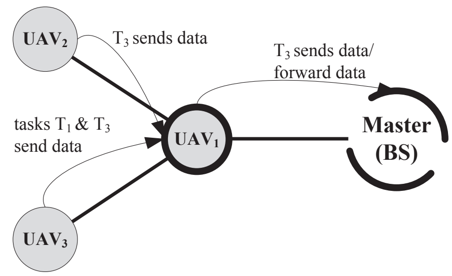

Fig. 10. Difficult behavior from $t_0 + \delta_2$ to $t_0 + \delta_3$

Instead, the traffic of both tasks is treated equally. But if both tasks are treated equal by a QoS management, QoS provisioning neither performs in a fair manner, nor performs efficiently regarding the utilized resources because both tasks must run with the higher priority of both tasks to ensure that both tasks perform fine. Thereby, the available data rate for concurrent data transmissions will be reduced and errors may occur. Besides, more resources will be used, especially energy consumption will be much higher as required by the tasks. If both tasks are run with the lower priority of both, energy consumption would be less, but also the required data rate may not be achieved and one of the tasks would not perform correctly anymore. The proposed approach on the basis of process-patterns allows to identify traffic with equal identification parameters because it considers the history of tasks. The autonomous QoS management handles the traffic generated by task T_3 with high priority because it knows the tasks and, thereby, the services that ran before. In this example, the Process-Pattern Manager knows that tasks T_1 and T_2 have been active before task T_3 became active.

The scheme also allows handling the problem that occurs when tasks T_1 and T_3 are active in parallel (see Fig. 10). The Process-Pattern Manager decides whether to use low QoS requirements of task T_1 or to use the high QoS requirements of task T_3. For this case, a fallback-mode is introduced: If a predecessor task has not finished and the pattern was triggered to transition to a following task equal to the running task, but with different QoS requirements, the Process-Pattern Manager works as usual, but compares the requirements of both tasks before. Until the former task has not finished, the stronger QoS requirements will be used for both flows to avoid negative influence towards the task with higher QoS requirements.

C. Discussion

The usage of groups within the communication structure is useful because often swarms of UAVs are performing the same tasks at the same time. As a result of using this structure, network control overhead can be reduced and the areal coverage can be extended in groups and, thereby, dependent of the purpose. Besides, energy consumption can be reduced and resources can be saved. A naive pattern-based approach would not be able to handle different QoS requirements of one service within different steps that cannot be differenced by their identification parameters. Using the proposed scheme in the whole network, an even-handed QoS provisioning, which depends on the execution context of each service and the environment in which the UAVs are deployed, can be implemented.

We developed a prototype and verified the described features within several test cases in a mesh network, but because no UAVs have been used for this setting, no general statement about the performance for certain types of UAVs is possible. From the test cases run with our prototype, we can deduce that the proposed scheme enhances the performance of the described fielding scenarios. This context-aware QoS management performs much more fair and demand-actuated than ordinary QoS management approaches for UAVs. Furthermore, the optimized control structure allows a much more efficient control of the UAVs while reducing energy consumption.

VI. Conclusion

QoS management is a problem in Wireless Mesh Networks environments, especially if these allow mobility and, thereby, dynamics in network topology. This problem is major in the case of wireless networks that consists of Unmanned Aerial Vehicles (UAVs). Most existing research focuses on improvements on ISO/OSI Layer 1/2 to increase performance. Considering the dynamic, process-pattern based behavior of UAVs, we introduced a process-pattern aware QoS control scheme. Furthermore, we used a hierarchical group communication structure that is based on each UAV's aerial position. Also this structure is controlled by the proposed scheme. The scheme improves the capabilities of a Wireless Mesh Networks consisting of UAVs. Our scheme allows the optimization of communication paths between UAVs and a central controller. In this scheme, the actual context in which the UAVs are working and performing their tasks builds the basis for the process-pattern aware QoS management. Our scheme works closer to real requirements on the level of QoS than just technical motivated QoS approaches. Future work is to validate the improvements in simulations of WNUAVs. Furthermore, we will include a business intelligence data aggregation mechanism on the basis of the process-pattern driven QoS control scheme. This should help to further optimize the occurring data flows in the WNUAV.

References

[1] K. Daniel, B. Dusza, A. Lewandowski, and C. Wietfeld, "Airshield: A system-of-systems muav remote sensing architecture for disaster response," in *Systems Conference, 2009 3rd Annual IEEE*, march 2009, pp. 196 –200.

[2] K. Daniel, S. Rohde, and C. Wietfeld, "Leveraging public wireless communication infrastructures for uav-based sensor networks," in *Technologies for Homeland Security (HST), 2010 IEEE International Conference on*, nov. 2010, pp. 179 –184.

[3] N. Goddemeier, K. Daniel, and C. Wietfeld, "Coverage evaluation of wireless networks for unmanned aerial systems," in *GLOBECOM Workshops (GC Wkshps), 2010 IEEE*, dec. 2010, pp. 1760 –1765.

[4] B. Dusza and C. Wietfeld, "Performance evaluation of ieee 802.16e mobile wimax for long distance control of uav swarms," in *Wireless Information Technology and Systems (ICWITS), 2010 IEEE International Conference on*, 28 2010-sept. 3 2010, pp. 1 –4.

[5] S. Rohde, N. Goddemeier, K. Daniel, and C. Wietfeld, "Link quality dependent mobility strategies for distributed aerial sensor networks," in *GLOBECOM Workshops (GC Wkshps), 2010 IEEE*, dec. 2010, pp. 1783 –1787.

[6] P.-B. Boek, D. Pielken, and Y. Tuechelmann, "Business process aware semantic qos provisioning scheme," in *Local Computer Networks (LCN), 2010 IEEE*, 2010, pp. 16 –23.

[7] W. Pattara-Atikom, P. Krishnamurthy, and S. Banerjee, "Comparison of distributed fair qos mechanisms in wireless lans," in *Global Telecommunications Conference, 2003. GLOBECOM '03. IEEE*, vol. 1, 1-5 2003, pp. 553 – 557 Vol.1.

[8] H. Luo, J. Cheng, and S. Lu, "Self-coordinating localized fair queueing in wireless ad hoc networks," *Mobile Computing, IEEE Transactions on*, vol. 3, no. 1, pp. 86 – 98, jan-feb 2004.

[9] Z. Kong, Y.-K. Kwok, and J. Wang, "Game theoretic packet scheduling to combat non-cooperativeness in wireless mesh networks," in *Distributed Computing Systems Workshops, 2008. ICDCS '08. 28th International Conference on*, 17-20 2008, pp. 162 –167.

[10] J. Lee, W. Liao, J.-M. Chen, and H.-H. Lee, "A practical qos solution to voice over ip in ieee 802.11 wlans," *Communications Magazine, IEEE*, vol. 47, no. 4, pp. 111 –117, april 2009.

[11] H. Zhai, J. Wang, and Y. Fang, "Distributed packet scheduling for multihop flows in ad hoc networks," in *Wireless Communications and Networking Conference, 2004. WCNC. 2004 IEEE*, vol. 2, 21-25 2004, pp. 1081 – 1086 Vol.2.

[12] C.-Y. Luo, N. Komuro, K. Takahashi, H. Kasai, H. Ueda, and T. Tsuboi, "Enhancing qos provision by priority scheduling with interference drop scheme in multi-hop ad hoc networks," in *Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008. IEEE*, nov. 2008, pp. 1 –5.

[13] H. Cheng and W. Zhuang, "Pareto optimal resource management for wireless mesh networks with qos assurance: Joint node clustering and subcarrier allocation," *Wireless Communications, IEEE Transactions on*, vol. 8, no. 3, pp. 1573 –1583, march 2009.

[14] H. T. Cheng and W. Zhuang, "Novel packet-level resource allocation with effective qos provisioning for wireless mesh networks," *Wireless Communications, IEEE Transactions on*, vol. 8, no. 2, pp. 694 –700, feb. 2009.

[15] W. Zhuang and H. T. Cheng, "An optimization framework for balancing throughput and fairness in wireless networks with qos support," *Wireless Communications, IEEE Transactions on*, vol. 7, no. 2, pp. 584 –593, february 2008.

[16] J.-S. Leu and H.-C. Hsieh, "An autonomous wireless sensor network with fault resilience," in *Networking and Communications, 2008. WIMOB '08. IEEE International Conference on Wireless and Mobile Computing,*, 12-14 2008, pp. 223 –227.

[17] S. Marwaha, J. Indulska, and M. Portmann, "Challenges and recent advances in qos provisioning in wireless mesh networks," in *Computer and Information Technology, 2008. CIT 2008. 8th IEEE International Conference on*, 8-11 2008, pp. 618 –623.

[18] C. Assi, A. Agarwal, and Y. Liu, "Enhanced per-flow admission control and qos provisioning in ieee 802.11e wireless lans," *Vehicular Technology, IEEE Transactions on*, vol. 57, no. 2, pp. 1077 –1088, march 2008.

[19] P.-B. Bök, D. Pielken, and Y. Tüchelmann, "Towards an integrated network control architecture," in *2010 Parallel and Distributed Computing and Networks Conference*. Acta Press, 2010.