Global Peer-to-Peer Classification in Mobile Ad-Hoc Networks: A Requirements Analysis

Dawud Gordon, Markus Scholz, Yong Ding, and Michael Beigl

Karlsruhe Institute of Technology, TecO, Karlsruhe, Germany
{firstname.lastname}@kit.edu
http://www.teco.kit.edu/

Abstract. This paper examines global context classification in peer-to-peer ad-hoc mobile wireless networks (P2P-MANETs). To begin, circumstances are presented in which such systems would be required to classify a global context. These circumstances are expounded upon by presenting concrete scenarios from which a set of requirements are derived. Using these requirements, related work is evaluated for applicability, indicating no adequate solutions. Algorithmic approaches are proposed, and analysis results in a benchmark as well as bounds for distribution of processing load, memory consumption and message passing in P2P-MANETs.

Keywords: Distributed Classification, Context Recognition, Peer-to-Peer, MANET, WSN, Requirements Analysis

1 Introduction

One of the advantages of pure P2P-MANETS[9] over structured client-server network architectures is their ability to adapt to new situations and account for mobility without drastically increasing complexity. The concepts of situational, context and activity recognition have been expanded to include ad-hoc mobile networks, such as wireless sensor nodes and cellular phones. In the ad-hoc network and embedded systems fields, these approaches have been focused on devices which are capable of recognizing their local situations and using this information for local decision making or communicating it to a centralized back-end system with various degrees of preprocessing, compression and data fusion.

These paradigms, while very useful for many applications, stand in contrast to the concept of P2P-MANETs. In embedded systems, local recognition by a device of its own situation can be very useful in local decision making processes. In distributed sensing systems, transmission of local situational information to a central location allows the system to recognize global situations and reduces the volume of communication when compared to forwarding unprocessed data.

For fully distributed ad-hoc wireless systems such as P2P-MANETs however, there is no theoretical, algorithmic or practical support available for global context recognition in related work. This paper will begin by identifying environments and example scenarios for global recognition in P2P-MANETs in Section 2, and extracting a list of requirements based on those scenarios in Section 3. In

Section 4, related work will be examined for applicable approaches, followed by a discussion in Section 5 and the conclusion in Section 6.

2 Application Scenario

Local situations refer to a situation occuring in the immediate environment of a network node, or subset of nodes, and can be sensed and recognized by that node or nodes. Global situations on the other hand, occur over the domain of the entire MANET and are not directly measurable at any one position, but are rather deduceable only when confronted with distributed measurements from multiple nodes within the network. The necessity to recognize global situations arise under the following circumstances:

The network has sporadic access to a data sink, network bridge or other communication module. A typical application for this kind of setting is provided by the landmarke ad-hoc firefighter support network[11]. This network is deployed in an environment with unstable communication channel characteristics[12]. However, despite a connection loss to the central uplink, the individual firefighter should still be informed about the situation of the entire team.

The network has access but at an exorbitant price in terms of energy consumption, bandwidth, delay, etc. An examplary use case for this setting could be free-range monitoring of livestock where only an expensive uplink is available to inform the care taker. While activities of individual animals are important, situations affecting the whole herd (e.g. herd fragmentation, being harried by a wolf, etc.) call for a global classifier (Fig. 1).

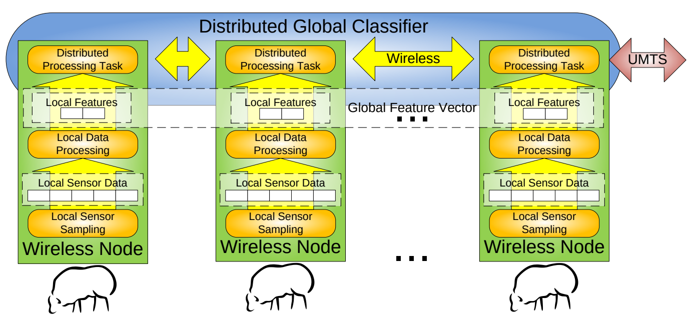

Fig. 1. Distributed Classifier Architecture in a P2P-MANET for Livestock Monitoring

The network without uplink must be situationally aware and act locally. Deploying P2P-MANETs in real life settings usually leads to incorrect function of the network due to unforseen permutations of environmental features. For instance, in the case of a distributed, autonomous sensor-actor system for monitoring and controlling a pump station, the observation of local network parameters could be the input to a distributed algorithm which could identify different types of failure situations and initiate preventative measures.

3 Requirements Analysis

Based on the application scenarios, it is possible to highlight the requirements which a distributed classification algorithm must fulfill:

Requirement 1 *Survival of Node Failures.* In all three scenarios it is clear that nodes may drop out of the network without warning due to connectivity issues or node failures. The global classifier algorithm must be able to continue functioning, even if in an impaired fashion, implying that any single point of failure within the network would break with this requirement.

Requirement 2 *Recovery from Node Failures.* Not only must the algorithm be able to survive failures, it is also crucial that it can recover from these failures, meaning that successive node replacements do not lead to long term degradation of the algorithm. In the livestock monitoring example, imagine a situation where there is a certain animal throughput, meaning animals are constantly being added to and removed from the herd. Without this capability the performance of the classification algorithm would slowly degrade over time as each animal which leaves the herd causes the irreparable loss of a certain amount of functionality.

Requirement 3 *Ability to Approximate the Mapping Function.* Each of the three scenarios represents a different mapping function from the input signals to the contextual ground truth. Moreover, in each scenario, the exact context which the system recognizes is only vaguely fixed and may be arbitrarily complex. As a result of this, an algorithm which would be able to accomplish these tasks must be able to learn the solution, no matter how complex the mapping function is.

4 Related Work

In parallel computing multiple nodes work simultaneously to reduce processing time compared to a sequential approach. A brief review of algorithms from this field showed that these either rely on a central coordinator [2], [10] or are managed by dedicated scheduling instances [7], failing to fulfill Req. 1 (survive node failures) or building on different conditions and cost models than MANETs.

Collaborative models and in-network data fusion are one of the most straightforward methods for P2P based classification. Therein each node contributes to a global consensus based on locally recognized situations. However, while approaches [14, 6] from this field employ different strategies to reach a global consensus, they are limited in the complexity of the mapping from local input to the global decision. These approaches observe the state of each of the nodes, and make a decision about the global situation based on these states, but without observing the identity or functionality of each node (voting). The global context algorithm is then only a function quantities of local contexts, in violation of Req. 3 as it can only map a subset of classification functions (see sec. 5 for a discussion). Recently [13] and [5] presented novel methods of processing context data within the nodes of a wireless network. However, there the classification is carried out by a single node, violating Req. 1 and 2. Finally, [8] presents a framework for distributed inference based on Bayesian networks and belief propagation. While this approach meets all requirements, "convergence may take a long time, or it may never happen" if the variables or the network are dynamic [8], making it inappropriate for situational recognition.

In Organic Computing, approaches such as swarm intelligence are distributed paradigms for solving optimization problems inspired by the biological processes of swarming, flocking and herding. Various authors from this field, e.g. [3], [1] present algorithms for the distributed detection and global classification of situations. However, these algorithms conduct this in a collaborative fashion which does not support Req. 3, or use a central unit to perform recognition over a feature map generated in a distributed fashion which is not reconcilable with Req. 1. In short, distributed classification approaches from the area of Organic Computing cannot be directly applied to global situational recognition in P2P-MANETs.

5 Analysis and Discussion

Social Role. In the machine monitoring example, all objects will have been present at classifier training time. This means that if a machine component is separated from the network and then later returns, the classifier will correctly map the data generated by that object to the output function of the classifier, as specified by Req. 1. The implication is that when the object is reconnected, whatever power was lost with respect to recognition at disconnect is regained.

In the case of livestock monitoring, animals may leave the herd and be replaced by other unique individuals. This presents the problem of how to include the new animals in classification. Simply substituting the data from the new animal in the global vector (see Figure 1) can also be problematic, as there is no reason to believe that the new animal plays the same role in the herd dynamics. Eventually, constant animal throughput would cause complete randomization of local vector locations in the global vector, leading to degradation of algorithmic performance, and situational inferences based on the previous data would no longer necessarily remain valid, violating Req. 2. This leads to the following:

Lemma 1. *A system in which new data is appended to the global vector at a position which is not role-dependent can be modeled by randomly re-locating the data from each node in the global vector at each classification phase.*

A possible way to combat this effect would be to train the classifier using random positions for each partial vector (the data generated by each node) from each object in the total feature vector. This assumes de facto homogeneity among the objects (e.g. livestock animals) as the information gathered from a certain animal can be input at any location on the feature vector without affecting the output of the classifier. Since no single object can assume a specific role, the only functions which can be mapped by the classifier at learning time are quantity-based functions (e.g. if the majority is sleeping then the herd is sleeping), rather than inferences based on the roles of certain individuals as to the situation of the whole (e.g. inferences based on the dominant roles of certain individuals). This yields the following:

Lemma 2. *A classifier trained on a global vector in which features from each object are appended to the global vector at random positions can only learn mappings based on quantities or counts of nodes.*

Unfortunately, functions over the quantities of objects reduces the system to majority and voting-based collaborative systems such as [14]. Standard classifiers implicitly learn object roles in the learning process as feature (vector) positions in the input vector are constant over time. A system in which these positions are not constant must therefore explicitly account for these fluctuations.

Theorem 1. *A global classifier which does not observe the individuality or role of each of the objects being monitored is only capable of classifying global contexts which can be reduced to functions over quantities and counts of node states.*

Brute Force Method. The simplest solution to the global classifier problem in terms of complexity is the brute force approach, in which each node transmits all locally generated data required for global context classification to every other node in the network, and then each node locally classifies the global situation. Theoretically, if the classifier is identical on each node, and the data vector is also identical, each node should locally classify the identical global situation.

The disadvantages include the amount of memory required by each node to store the entire classifier, the number of transmissions required to transmit all data generated to every other node, as well as energy consumption due to the redundancy. On the other hand, the network is extremely stable as failed nodes do not adversely affect the classification of the rest of the network, as long as the classifier used can accommodate the variable feature vector length (see [4]). Also, new nodes which are added to the network must only receive the parameters for the classifier and be added to the global list of data publishers and subscribers in order to become functioning members of the new system.

Other Solutions One approach would be to select a classification algorithm which easily lends itself to distributed execution and apply this to the entire network. Such algorithms are often referred to as connectionistic methods, (e.g. neural network, multi-agent system, spatial reasoning, etc.) which involve processors (neurons) and connections between these processors. This would reduce processor load and memory required when compared to the brute force approach, though it is initially unclear what affect this would have on communication between nodes. Such a method requires time synchronization which is indeed costly in ad-hoc P2P networks, though it would overcome the convergence issues of [8], and increased communication could possibly be combated by P2P self-organization.

Another approach would be to distribute the data instead of the execution. This could be accomplished by adapting instance-based learning methods such as k-Nearest-Neighbors or Self Organizing Maps to be distributed over multiple nodes along the principle that vectors which are close to each other are also close to each other in terms of hops. Once again, self-organization could be employed to account for varying network structure and mobility, but the amount of communication incurred and the advantages over brute force must be studied.

Resource Consumption Analysis Assuming N peer-to-peer nodes and objects in the network, and a distributable global classification algorithm with memory consumption M and processing load P. The brute force approach incurs the full memory consumption of M and processing load P locally at each node, as the classifier is redundantly stored and executed. The number of messages which have to be passed between nodes is $N-1$, as each node needs to communicate local features to every other node in order to build the global feature vector, or $N(N-1)$ messages in total. The memory consumption is thereby increased to $(M+S_g)$, where S_g is the size (length) of the global feature vector.

For a distributed connectionist reasoning approach, assuming each node is an input, output and hidden processor (e.g. neuron), then each node will have to pass 2 messages. Each processor requires input and generates output, where the input for the input processors is generated locally, and the output processor is output locally. In other words, per classification phase $2N$ messages must be passed by the system. Local memory consumption is now that incurred by 3 of $3N$ processors, where $3N$ processors can be held in M memory, or $\frac{M}{N}$, plus the length of the local feature vector, giving $\frac{M}{N}+S_l$. Each node must execute 3 of $3N$ processors, where the total processing load is P, yielding a load of $\frac{P}{N}$ per node. This indicates that this approach would reduce memory consumption by $\frac{M(N-1)}{N}+(S_g-S_l)$, processing load by $\frac{P(N-1)}{N}$ and the number of messages passed by $N(N-3)$.

Taking this one step further, we can hypothesize about the lower bounds for resource consumption in P2P-MANETs. In an optimal situation, each node sends local information to the exact logical location where it is needed (1 hop), and the system has no redundancy, indicating that each node transmits 1 message per classification phase, for a total of N messages. Also, optimally the system would distribute the memory consumption M and processor load incurred P equally across all nodes, yielding $\frac{M}{N}$ and $\frac{P}{N}$ respectively. This indicates that while being optimal in terms of memory and processor requirements, the connectionist reasoning approach would still be sub-optimal in terms of message passing by a factor of 2.

6 Conclusion

This work began by identifying the need for peer-to-peer classification of global situations in MANETs. This is based on three different circumstances which occur in a subset of standard deployments, where either there is no communication with the outside world, that communication is very expensive, or a link is only available from time to time. These situations were elaborated on by presenting three example scenarios, safety monitoring in firefighter teams, monitoring and alerts in livestock management, and industrial monitoring and controlling. These scenarios were then analyzed in order to extract requirements for a peer-to-peer classification algorithm in wireless ad-hoc networks. This analysis indicated a further requirement of respecting heterogeneity of the different objects being monitored. Hypothetical upper and lower bounds for processing load, memory usage and communication volumes were elaborated, and a brute force (upper bound) and neural network (close to lower bound) approach were examined.

The authors acknowledge joint funding by the DFG under the project "Sense-Cast", the EC under "CHOSeN" and the BMBF under "Koordinator".

References

1. Sven A. Brueckner, H. Van, and Dyke Parunak. Swarming distributed pattern detection and classification. In *Lecture Notes in Computer Science*, pages 232–245. Springer Verlag, 2005.
2. Doina Caragea, Adrian Silvescu, and Vasant Honavar. A framework for learning from distributed data using sufficient statistics and its application to learning decision trees. *Int. J. Hybrid Intell. Syst.*, 1:80–89, April 2004.
3. P. Dasgupta. A multiagent swarming system for distributed automatic target recognition using unmanned aerial vehicles. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 38(3):549 –563, May 2008.
4. Dawud Gordon, Stephan Sigg, Yong Ding, and Michael Beigl. Using prediction to conserve energy in recognition on mobile devices. In *9th International Conference on Pervasive Computing and Communications (PERCOM Workshops)*, 2011.
5. Guang Lu and Wei Xue. Adaptive weighted fusion algorithm for monitoring system of forest fire based on wireless sensor networks. *Computer Modeling and Simulation, International Conference on*, 4:414–417, 2010.
6. Ping Luo, Hui Xiong, Kevin Lü, and Zhongzhi Shi. Distributed classification in peer-to-peer networks. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '07, pages 968–976, New York, NY, USA, 2007. ACM.
7. Ramesh Natarajan, Radu Sion, and Thomas Phan. A grid-based approach for enterprise-scale data mining. *Future Gener. Comput. Syst.*, 23:48–54, 2007.
8. Mark A. Paskin and Carlos E. Guestrin. Robust probabilistic inference in distributed systems. In *Proceedings of the 20th conference on Uncertainty in artificial intelligence*, UAI '04, pages 436–445, Arlington, Virginia, 2004. AUAI Press.
9. R. Schollmeier. A definition of peer-to-peer networking for the classification of peer-to-peer architectures and applications. In *Peer-to-Peer Computing, 2001. Proceedings. First International Conference on*, pages 101 –102, August 2001.
10. Markus Scholz, Gesine Flehmig, Hedda R. Schmidtke, and Gerhard H. Scholz. Powering smart home intelligence using existing entertainment systems. In *the 7th International Conference on Intelligent Environments (IE'11)*, 2011.
11. Markus Scholz, Till Riedel, and Christian Decker. A flexible architecture for a robust indoor navigation support device for firefighters. In *Proceedings of the 7th International Conference on Networked Sensing Systems*, 2010.
12. Erhard Schubert and Markus Scholz. Evaluation of wireless sensor technologies in a firefighting environment. In *Proceedings of the 7th International Conference on Networked Sensing Systems*, 2010.
13. Stephan Sigg and Michael Beigl. Expectation aware in-network context processing. In *Proceedings of the 4th ACM International Workshop on Context-Awareness for Self-Managing Systems*, CASEMANS '10, New York, NY, 2010. ACM.
14. Georg Wittenburg, Norman Dziengel, Christian Wartenburger, and Jochen Schiller. A system for distributed event detection in wireless sensor networks. In *IPSN '10: Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks*, New York, NY, 2010. ACM.