Ad-hoc Routing Metrics and Applied Weighting for QoS support

Migas, Nikos and Buchanan W. J.
School of Computing, Napier University, EH10 5DT, Scotland, UK
{n.migas, w.buchanan}@napier.ac.uk

Abstract

In the vast majority of ad-hoc routing protocols, the hop-counting mechanisms for identifying the optimal route are dominant. However, this approach oversimplifies such a complex decision by ignoring the fact that participating devices may have considerably unequal performance characteristics and current utilisation status. Accordingly, it is possible for an optimal route to be composed of devices with high utilisation status, or, low battery reserves, which results in an overall unreliable route. This research work tackles this by identifying the best metrics that can describe any route within a graph, in terms of overall throughput, reliability, and minimum energy consumption. Simulations were carried out by varying critical factors of mobile devices such as battery reserves, memory and CPU utilisation, and results recorded the effect that this has on the device's overall routing metric. This paper also presents the threshold values, which turn the device from routing-capable to routing-incapable state.

1. Introduction

Recent advances in technology introduced a wide range of devices with different performance characteristics, and the ability to wirelessly communicate with one another without the need of a fixed infrastructure. Networks of this category are commonly known as ad-hoc networks, which can be generally defined as a collection of geographically distributed nodes that communicate in a multi-hop fashion and are responsible for location management and data routing [16], [9]. Mobility is usually a prime feature of ad-hoc networks, and thus provides the ability to users to communicate, cooperate, and access the Internet services in an *any-time-anywhere-fashion* [6]. Ad-hoc networks have been proposed as a networking solution for those situations where the network set-up time is a major constraint, or, a network infrastructure is either not available, or, not desirable [15]. Applications of ad-hoc networks can be originally found in military, rescue, and antiterrorism operations, whereas, some commercial ones include: conferencing; sensor networks; personal area networks; and embedded computing applications [14].

Despite the theoretically enormous benefits of ad-hoc networks, there are certain constraints that limit the implementation of this technology to its full potential. Routing is the most challenging issue, as participating devices are responsible for deriving and maintaining routing information, and actually route data packets, as there is no fixed infrastructure. Furthermore, as the majority of these devices are mobile, the network becomes highly dynamic, and routes that may be considered optimal, at a given time, may suddenly become unavailable, or, undesirable. In addition, mobile devices rely on battery power, which exhausts at a faster rate while the device has its wireless features on, or, it is routing data packets [11]. An extensive survey of routing techniques that can be applied to conserve battery life is presented in [8].

This research work attempts to address these issues by assigning a fitness metric to each participating device, which can then be integrated and used by the routing protocol to make precise judgments on which paths can provide the desired (Quality of Service) QoS guarantees for certain types of network traffic. Section 2 presents the domain of this research work. Section 3 presents the process of calculating the metrics of individual devices and the overall metrics that correspond to the paths that these devices create. Section 4 presents results concerning the capability of different device types to route certain types of traffic, and the effect that variation of critical device elements may have on this determination. Section 5 concludes the paper.

2. Domain

There are many different ad-hoc routing protocols, proposed in the literature, to date. Generally, these can be categorised by the way they maintain routing information. For example, routing protocols, which maintain routing information for each node on the network, at all times, are categorised as proactive, while protocols which discover routing information only when it is required, can be categorised as reactive. The reader may refer to [1] for an extensive survey on routing protocols and their functions.

The common ground, that these approaches share, is the mechanism for deciding upon *optimal* routes, and, the lack of support for multiple redundant paths. The hop-counting mechanism, which is typically employed for *optimal* route determination, can significantly reduce the reliability, and, performance of the network. With this, a route is represented by the number of intermediate nodes, which needs to be traversed to reach the destination. For example, a route with three intermediate nodes is considered to be fitter than a route with more than three. However, this is not always true, especially with the vast diversity of performance capabilities of mobile devices. In addition, different routing

978-1-4244-1694-3/08/$25.00 ©2008 IEEE

scenarios impose different requirements, and thus a path may be suitable for a certain objective but unsuitable for another. The determination of the *optimal* route is a complex task, and requires research into the *best* metrics, such as memory capacity, network performance, processing capabilities, and so on, which should also be appraised to the routing objective that the route is seeking to accomplish [10].

2.1. Metric-driven routing for QoS support

A number of innovative methods have been proposed in the literature, which aim to support path redundancy, and QoS, while maintaining low network overhead. As an example, [12], proposed a new protocol called Disjoint Path Selection Protocol (DPSP) that supports communication between networked nodes over multiple diverse paths. They showed that DPSP increases reliability, and, provides support for QoS-driven applications. A novel QoS-aware resource discovery framework for ad-hoc networks has also been proposed by [9]. In their framework, distributed self-organising discovery agents are responsible for monitoring QoS information, and are used to predict path QoS on behalf of other nodes. Preliminary simulation results showed that their framework enhances QoS-awareness, when compared to traditional centralised approaches. Furthermore, an innovative routing scheme called trigger-based distributed routing (TDR) for supporting real-time QoS traffic in mobile ad-hoc networks (MANETs) was proposed by [5]. That scheme uses failure prediction-based alternate route discovery, which avoids the need of maintaining unnecessary routes, and thus reduces control traffic, and the size of nodal databases. Simulation results proved that TDR provides significant improvements compared to *prediction-less* QoS routing protocols such as E-AODV [13], and DQoSR [4]. Additional research work in the same area can be found in [2], [7], [17].

Even though these methods are shown to improve over traditional techniques, they take into little, or, no consideration, key metrics, such as nodal computation strength, utilisation status, and battery reserves. In an attempt to prove the need for those parameters to be included in the route determination process, the authors in [3] showed that wireless devices not only have huge performance differences in processing speed, but also in network transmission reliability. In addition, they showed that routing of heavy network traffic imposes higher resource-consumptions in terms of CPU, memory, and battery discharge rate to resource-constrained devices, than any other device of a fitter category.

3. Model

This section describes the process of assigning a routing metric to an ad-hoc routing device, which is based on the results that are produced by the test agents in [3]. Initially, for each test that participates in the metric calculation process, a preliminary metric is calculated, which is then appropriately weighted to suit various routing objectives, and is then averaged with the remaining weighted preliminary metrics. In this manner, a number of overall routing metrics is calculated, which represent the routing fitness of a device to achieve various routing objectives, such as to route synchronous network traffic, asynchronous network traffic, and so on. The nodes' overall routing metrics are gathered by the protocol’s on-demand route discovery, or, proactive network discovery process. This information is used by a source node to determine the capability/incapability of each retrieved route to accomplish the source's routing scenario. Thus, the source estimates a final metric for each of the capable routes, which represents the QoS that the routes can offer, and it bases its route selection on the requirements imposed by its routing scenario. The tests that participate in the preliminary metrics calculation process are outlined in Table 1.

Table 1: Tests outline and representation

Test	Symbol
1D Bubble sort	T_1
Merge test	T_2
Storage & memory 1 File	T_3
Storage & memory 1 KB	T_4
Client – Server throughput 1m	T_5
Proxy throughput 1m	T_6
TCP error	T_7
IP error	T_8
UDP error	T_9
CPU utilisation	T_{10}
Memory utilisation	T_{11}
Battery level	T_{12}

3.1. Preliminary metric calculation

For each of the tests presented in Table 1, a preliminary metric (pm) is calculated, based on the results achieved by each test. The process of creating the pm(s) is based on either a function, or, a threshold value:

- **Function**. Results acquired from a certain test are passed to a function which produces a preliminary metric.
- **Threshold value (TH)**. It represents the worst case scenario for a test.

The preliminary metrics for tests T_1 - T_6 are calculated based on a threshold basis while tests T_7 - T_{12} are calculated using functions. Table 2 presents the default threshold values for some of these tests.

Table 2: Threshold values for preliminary metric calculation

Test	Threshold value (TH)

1D Bubble sort	500
CPU Merge	100
memory 1 File	20
memory 1 KB	7000
Client-Server throughput	80
Proxy throughput	350

The threshold values were derived from experimentation (*see* [3]), and represent the worst case test results, that an ad-hoc routing device can achieve. A device which achieves results equal to a threshold, or above, is determined incapable of routing, and is thus assigned an infinity routing metric (∞). The mathematical expression that is used to estimate the preliminary metric (PM), given the values of the test results (T_n) and threshold value (TH_n) for that test is:

$$PM = \left(\frac{T_n}{TH_n}\right) \times 100 \tag{1}$$

The preliminary metrics for tests $T_7 - T_9$ are calculated using the mathematical expressions in:

$$PM_7 = \frac{TCP_{error}}{TCP_{in} + TCP_{out}} \times 100 \tag{2}$$

$$PM_8 = \frac{IP_{error}}{IP_{in} + IP_{out}} \times 100 \tag{3}$$

$$PM_9 = \frac{UDP_{error}}{UDP_{in} + UDP_{out}} \times 100 \tag{4}$$

For T_{10} - T_{11}, the *pms* are calculated based on equation 5, while for T_{12} is based on equation 6. In both cases, the special sensitivity factors α, β, and γ are introduced. These factors differ on value for each of these tests, and thus allow for better adaptation of the output preliminary metrics accordingly. In particular, these factors refine the shape of the exponential curves (*see* equation 5-6), and their values were deduced through simulation, aiming to deliver the precise preliminary metric for each memory, CPU, and battery reading, respectively. For example, the equation in 6 in relation to the default sensitivity factors in Table 3, produce a low preliminary metric for battery readings above 60% of battery reserves, while for readings below 20% they produce a high preliminary metric. Although the default values of the sensitivity factors might not constitute the most accurate figures, they were shown to produce close proximity preliminary metrics.

$$f(x) = 100 \times e^{-\alpha \cdot \left(\frac{100-x}{100}\right)^{\beta} + \gamma} \tag{5}$$

$$f(x) = 100 \times e^{-\alpha \cdot \left(\frac{x}{100}\right)^{\beta} + \gamma} \tag{6}$$

Table 3: Sensitivity factors for each monitoring test

	α	β	γ
CPU utilisation	3	2	-4
Memory usage	3	2.5	-4
Battery level	6	2	0

3.2. Overall metric calculation

Once the preliminary metrics for all tests have been calculated, the system applies a distinct weighting (*W*) to each one of them according to various objectives and calculates an overall metric for each of the objectives. The mathematical expression used to calculate the overall metric (OM) is:

$$\text{OM} = \frac{\sum_{n=1}^{n=12} (W_n \times PM_n)}{\sum_{n=1}^{n=12} (W_n)} \tag{7}$$

The system supports various objectives including:

- **Energy efficient traffic (O_1)**. This type of traffic typically favours devices with high battery levels.
- **Synchronous traffic (O_2)**. It requires high throughput and good buffering capabilities.
- **Asynchronous traffic (O_3)**. This type of traffic typically has no special requirements.
- **Critical traffic (O_4)**. It requires reliable transmission to the destination.
- **Secure traffic (O_5)**. Typically requires encryption/decryption, and thus good processing capacity.
- **Burst traffic (O_6)**. This type of traffic has high buffering requirements.

In addition to these, the routing protocol can dynamically adapt to newly defined objectives, as long as their specification concerning their weighting requirements is provided.

The weighting system is different for each of these objectives, and thus each of these is treated differently, that is, according to the requirements. In this way, more weighting can be applied to battery preliminary metric for energy efficient traffic, while less can be applied for asynchronous traffic. The values of the weighting system were deduced through experimental work, and were shown to produce the desired outcome through simulations (*see* Section 4).

The final stage includes the translation of the devices' overall metric along a source route, to a meaningful expression, which indicates the ability of the route to accomplish the objective in question. For this purpose, five grades were defined including: excellent, very good, good, average, and poor.

Figure 2 illustrates the process of assigning an overall routing metric to an ad-hoc routing device based on a number of pre-defined objectives, as well as calculating the final metric of the route in which the device is situated. Initially, when a source node receives the preliminary metrics of each node along a source route, which is achieved by initiating a route discovery process, it calculates the capability/incapability determination of each node along the source route, in relation to the intended routing objective, and based on this information it calculates the capability/incapability of the source route. Table 4 provides a look-up table on which a node is based to determine the capability/incapability of each node along a source route, as it defines the desired overall metric ranges, in which a device must fall to be determined as capable. These values were deduced through experimentation, and were verified through simulations (*see* Section 4).

Table 5 is used as a look-up table by a source node in order to translate the capable retrieved routes into a final metric. The average (AV) of the devices' overall routing metrics, as well as the standard deviation (SD) of these metrics, is used to determine the QoS-level that a route can provide. The average represents the routing fitness of the route, while the standard deviation represents the difference in routing fitness that the nodes along a route may have. Thus, a route with low average and standard deviation is likely to provide high-levels of QoS, and is thus preferable for routing objectives that impose high requirements.

Table 4: Desired ranges for each predefined objective

Objective	Desired metric ranges
Energy efficient network traffic	0-30
Synchronous network traffic	0-25
Asynchronous network traffic	0-50
Critical network traffic	0-20
Secure network traffic	0-15
Burst network traffic	0-20

4. Results

In order to demonstrate the routing metric calculation process, six distinct device types have been defined:

- **Average strength iPAQ PDA (DT_1)**. Most common in mobile ad-hoc networks.
- **iPAQ PDA with high utilisation (DT_2)**. Same as above but with high CPU utilisation.
- **iPAQ PDA with good network throughput (DT_3)**. Same as first category, however, the proxy throughput was set to a valid maximum for the standards of these devices.
- **iPAQ PDA with poor throughput (DT_4)**. These devices provide a valid minimum proxy throughput.
- **iPAQ PDA with high errors in the network protocols (DT_5)**. These devices are prone to network protocol errors.
- **iPAQ PDA with low battery (DT_6)**. These devices were set to have low battery capacity.
- **Average strength laptop (DT_7)**.
- **Good strength laptop (DT_8)**.
- **Powerful workstation (DT_9)**. An exceptionally strong device, which is not battery-driven.

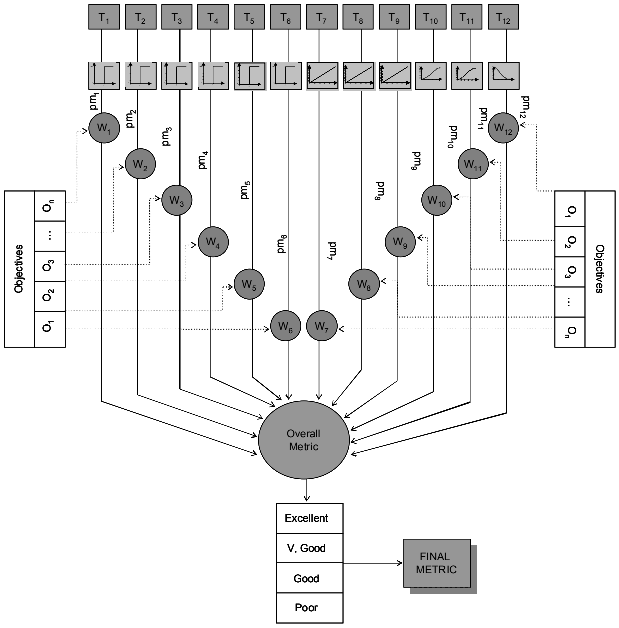

Figure 2: The calculation of the overall routing metric.

The experimentation results presented in [3], and [11], are being recapitulated and further enhanced in order to support the additional device types defined in this paper. Table 6 presents the test results, which have been achieved, as presented in [3], along with the conditions and implementation details. Based on the mathematical equations 1-6, and on test results presented on Table 6, the preliminary metrics for each device type can be calculated, as shown in Table 7. The next step involves the determination of the device's capability/incapability of accomplishing certain routing scenarios. Based on the desired ranges of the six previously defined objectives (*see* Table 4) and on the overall metrics (*see* Table 8), the capability/incapability determination can be calculated, as shown in Table 9.

Accordingly, the average PDA is classified as capable of routing energy efficient network traffic, as its memory, CPU, and battery preliminary metrics are low. However, it falls outside the limits of synchronous traffic requirements, as its network throughput was not within the required limits. For asynchronous traffic, an average PDA is sufficient to route data, as this type of traffic does not have any special requirements, apart from the battery metric which is always a key metric in all of the objectives. The average PDA is also capable of routing critical traffic, as its network protocol error rates are significantly low. Finally, it is classified as incapable of routing secure traffic or burst traffic, as it has low buffering capabilities and normally takes considerable time to perform intensive calculations.

4.1. Preliminary Metric Simulations

Simulations were conducted based on a number of device/objectivity combination by varying a number of key preliminary metrics, including: CPU utilisation, memory utilisation, and battery level. For example, the CPU utilisation of a device can easily decrease or increase according to the user's actions, such as the user has started a resource-consuming application, which increased the CPU utilisation by 35%. Similarly, the amount of free memory available to the system can change for the same reasons. The battery, a metric of vast significance, varies with time and type of usage. For example, if a resource-constrained device is used as an ad-hoc router, it will cause the battery to decrease at a rate of approximately 30% faster than if it was idle [11].

Figure 3 presents the variation of the overall metric when the average PDA's CPU preliminary metric ranges from zero to 100. According to Table 10, if the CPU preliminary metric exceeds the value of 40.43, the PDA becomes incapable of routing energy efficient network traffic. This is due to the fact that increased CPU utilisation can cause the battery to decrease at a much faster rate. In contrast, the CPU increase does not cause the overall metric to exceed the upper limit for asynchronous traffic, and thus the PDA will always be capable of routing this type of traffic, irrespective of the current CPU utilisation. However, if the CPU preliminary metric exceeds the predefined value of 95.25, the system automatically detects this and sets the overall metric to point to infinity. This is not illustrated in the graphs throughout this section for presentation purposes. Finally, the PDA turns to the incapable state for critical traffic after the CPU preliminary metric exceeds the threshold value of 35.03. Table 10 summarises the threshold values for each simulated preliminary metric, which can cause an average PDA to inverse its capability state for objectives O_1, O_3 and O_4, while Table 11 maps these threshold values to the actual corresponding CPU utilisation, memory usage, and battery level.

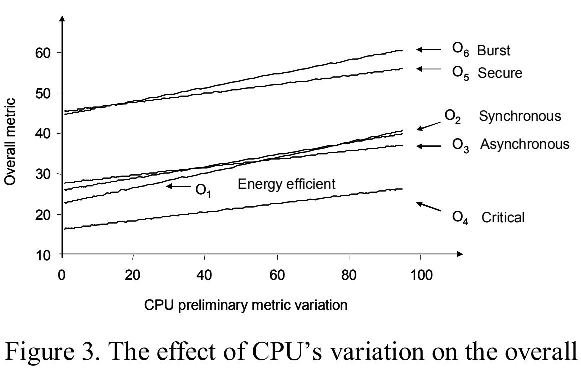

Figure 3. The effect of CPU's variation on the overall metric (average PDA)

Table 10: Threshold values for each simulated preliminary metric (av. PDA)

	CPU	Memory	Battery
O_1	Metric ≥ 40.43	Metric ≥ 91.96	Metric ≥ 21
O_3	Metric ≥ 95.25	Metric ≥ 95.83	Metric ≥ 87.37
O_4	Metric ≥ 35.03	Metric ≥ 91.38	Metric ≥ 17.38

Figure 4 presents the variation in the overall metric for the PDA with high CPU utilisation device type, where the battery preliminary metric was variable. It can be seen that the battery level can have a strong impact on the overall metric, especially when the PDA's CPU is heavily utilised. Thus, as the CPU being overutilised, the PDA can only route asynchronous traffic (*see* Table 12). However, if the battery exceeds the threshold values of 66.65 the PDA moves to the incapable state for this type of traffic. As constant high CPU utilisation causes the battery to discharge at considerably faster rate than low utilisation [11], it is important to exclude it from routing when the remaining battery approaches low levels.

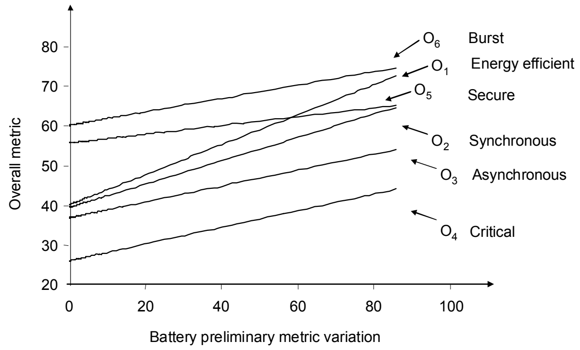

Figure 4: The effect of battery's variation on the overall metric for the PDA with high CPU utilisation

Table 12: Threshold values for each simulated preliminary metric for the PDA with high CPU utilisation

	CPU	Memory	Battery
Asynchronous	Metric ≥ 95.25	Metric ≥ 95.83	Metric ≥ 66.65

Figure 5 presents the variation in the overall metric for the PDA with a poor throughput device type, where the battery preliminary metrics was variable. This shows that a PDA with poor throughput can still maintain the ability to achieve the same objectives as an average PDA, however, for the energy efficient objective, heavier restrictions are imposed in terms of CPU utilisation, memory usage, and battery capacity. According to Table 13, the threshold value of the CPU preliminary metric is reduced by 13 units, and approximately eight units for the memory and battery. The reason for this is that a PDA with low throughput consequently requires more time to route data, and thus more battery is consumed throughout this process, and also the CPU and memory is utilised for a longer duration.

Figure 6 presents the variation in the overall metric for the PDA with high error network protocol rate device type, where the battery metric was variable. As shown in Table 14, this device type can only support two objectives: energy efficient and asynchronous.

5. Conclusions

This research work extends the study presented in [3] and [11] by adding support for the overall metric calculation of a routing device. The value of this metric is dependent on the objective the device is seeking to accomplish. In this paper, six distinct objectives and nine device types were defined, all with different requirements and characteristics respectively. It has been shown that the metric calculation process is correct, as each device type is assigned a distinct metric for each objective, and determined as capable or incapable according to the desired configuration. In this way, QoS can be guaranteed, as each device will always be assigned to certain routing scenarios according to its capabilities, utilisation, and network status. In addition, this method allows *low-requirements* network traffic to flow through *non-optimal* routes, and therefore optimal routes may not be overburdened.

Furthermore, a number of simulation cases were presented in order to demonstrate the effect that changes of vital device elements can have on the overall metric. Results show that when key metrics are changed, such as the remaining battery drops, or that the CPU is highly utilised, or that the device is running low on available memory, the device turns to the incapable state of routing *high-requirements* traffic types. The variation of the overall metric is adequately sensitive in all cases, and thus this demonstrates the ability of the proposed scheme to rapidly respond to critical changes. In addition, the threshold values, when a device becomes incapable of achieving a certain objective, are presented and are fully justified.

We are currently focusing in extending our work to simulate a number of routing scenarios in order to show whether this on-demand metric-driven approach can also significantly reduce network traffic and latency.

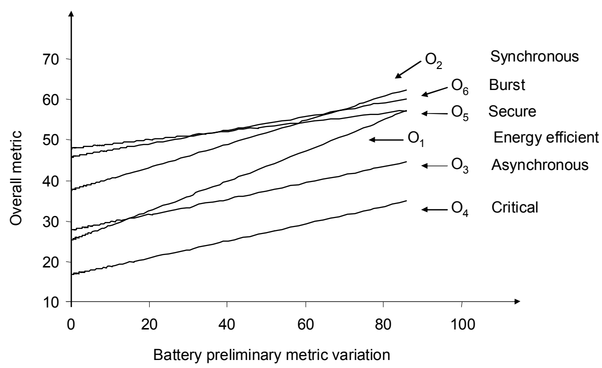

Figure 5: The effect of battery's variation on the overall metric for the PDA with poor throughput

Table 13: Threshold values for each simulated preliminary metric for the PDA with poor throughput

	CPU	Memory	Battery
Energy efficient	Metric ≥ 27.56	Metric ≥ 83.29	Metric ≥ 13.28
Asynchronous	Metric ≥ 95.25	Metric ≥ 95.83	Metric ≥ 87.37
Critical	Metric ≥ 32.45	Metric ≥ 93.82	Metric ≥ 16.28

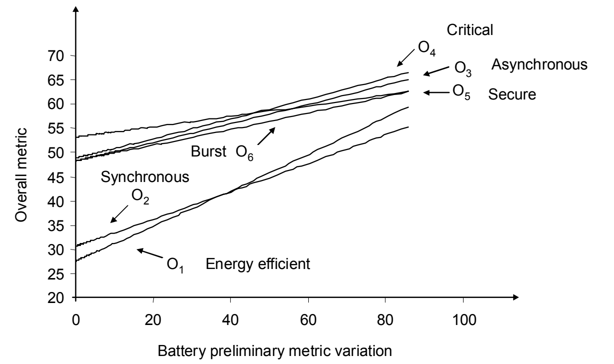

Figure 6: The effect of battery's variation on the overall metric for the PDA with high error rate

Table 14: Threshold values for each simulated preliminary metric for the PDA with high error rate

	CPU	Memory	Battery
Energy efficient	Metric ≥ 15.34	Metric ≥ 75.19	Metric ≥ 7.32
Asynchronous	Metric ≥ 19.97	Metric ≥ 69.81	Metric ≥ 9.24

9. References

[1] A review of routing protocols for mobile ad hoc networks. Ad-hoc networks. Vol. 2. pp. 1-22.

[2] A Simulation Analysis on reactive Route Repair techniques for QoS sensitive applications in Mobile Ad Hoc networks. MobiHoc '00. Boston, Massachusetts. pp. 139-140.

[3] Analysis of an Agent-based Metric-Driven for Ah-hoc, On-Demand Routing. Ad Hoc Networks. In Press, Corrected Proof. Available online 3 July 2004.

[4] Distributed Quality-of-Service Routing in Ad Hoc Networks. IEEE Journal on Selected Areas in Communications. Vol. 17, No. 8. pp. 1488-1505.

[5] Trigger-Based Distributed QoS Routing in Mobile Ad hoc Networks. SIGMOBILE Mobile Computing and Communications Review. Vol. 6, No. 3. pp. 22-35.

[6] Wireless ad-hoc networking – The art of networking with-out a network. Ericsson Review. No. 4. Available from <http://www.ericsson.com/about/publications/review/2000_04/files/2000046.pdf>

[7] Securing Quality-of-Service Route Discovery in On-

Demand Routing for Ad Hoc Networks. SASN '04: Proceedings of the 2nd ACM workshop on Security of ad hoc and sensor networks. Washington DC, USA. pp. 106-117.

[8] A survey of energy efficient network protocols for wireless networks. Wireless Networks. Vol. 7, No. 4, pp. 343-358.

[9] A Unified Framework for Resource Discovery and QoS-Aware Provider Selection in Ad Hoc Networks. SIGMOBILE Mobile Computing and Communications Review. Vol. 6, No. 1. pp. 13-21.

[10] Mobile Agents for Routing, Topology Discovery, and Automatic Network Reconfiguration in Ad-Hoc Networks. 10th IEEE International Conference and Workshop, ECBS Huntsville, USA, pp. 200-206.

[11] Metric Evaluation of Embedded Java-Based Proxies on Handheld Devices in Cluster-Based Ad Hoc Routing. 12th IEEE International Conference and Workshops on the Engineering of Computer-Based Systems (ECBS'05). Washington D.C., USA. pp. 147-154.

[12] Path Set Selection in Mobile Ad Hoc Networks. MobiHoc '02: Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing. Lausanne, Switzerland. pp. 1-11.

[13] Quality of Service for Ad Hoc On-Demand Distance Vector Routing. IETF Internet draft. draft-ietf-manet-aodvqos-00.txt. Work in progress.

[14] Ad-hoc networking: an introduction. Ad-hoc networking. Pub-lished by Addison-Wesley. ISBN: 0201309769

[15] Performance Analysis of TCP over static Ad-Hoc Wireless Networks. In Proceedings of the ISCA 15th International Conference on Parallel and Distributed Computing Systems (PDCS). Louisville, USA. pp. 410-415.

[16] A Multicast Routing Algorithm Based on Mobile Multicast Agents in Ad-Hoc Networks. Special Issue on Internet Technology, IEICE TRANS. COMMUN. Vol. E84-B, No. 8. pp. 2087-2094.

[17] Dynamic QoS Allocation for Multimedia Ad Hoc Wireless Networks. Mobile Networks and Applications. Vol. 6, No. 4. pp. 377-384.

Table 5: Final route metric look-up table

AV & SD	Energy	Synch	Asynch	Critical	Secure	Burst
AV≤3 & SD=0	Excellent	Excellent	Excellent	Excellent	Excellent	Excellent
AV≤5 & SD=0	Excellent	Excellent	Excellent	Excellent	V.Good	Excellent
AV≤5 & SD≤4	Excellent	V.Good	Excellent	Excellent	V.Good	V.Good
AV≤5 & SD≤5	Excellent	V.Good	Excellent	Excellent	Good	Good
AV≤10 & SD≤4	Excellent	V.Good	Excellent	V.Good	Good	V.Good
AV≤10 & SD≤5	Excellent	V.Good	Excellent	V.Good	Good	Good
AV≤10 & SD≤8	V.Good	Good	Excellent	V.Good	Good	Good
AV≤10 & SD≤10	V.Good	Good	Excellent	V.Good	Poor	Poor
AV≤15 & SD≤9	V.Good	Good	Good	Good	Poor	Good
AV≤15 & SD≤10	V.Good	Good	Good	Good	Poor	Poor
AV≤15 & SD≤15	V.Good	Poor	Good	Good	Poor	Poor
AV≤20 & SD≤15	Good	Poor	Good	Poor	Poor	Poor
AV≤30 & SD≤20	Poor	Poor	Good	Poor	Poor	Poor
AV≥30 & SD≥0	Poor	Poor	Poor	Poor	Poor	Poor

Table 6: Test results for each device type

DT_1	DT_2	DT_3	DT_4	DT_5	DT_6	DT_7	DT_8	DT_9	DT_1
31.83	31.83	31.83	31.83	31.83	31.83	2.61	1.395	0.56	31.83
76.34	76.34	76.34	76.34	76.34	76.34	7.38	1.69	0.74	76.34
16.99	16.99	16.99	16.99	16.99	16.99	2.58	0.27	0.24	16.99
5313.7	5313.7	5313.7	5313.7	5313.7	5313.7	1127	279.3	196	5313.7
0	0	0	0	70	0	0	0	0	0
0	0	0	0	70	0	0	0	0	0
20	20	20	20	85	20	0	0	0	20
52.544	52.544	52.544	52.544	52.544	52.544	44.52	38.104	28.32	52.544
80.5	80.5	14	299.985	80.5	80.5	5.25	2.625	1.75	80.5
3	89	3	3	3	3	2	2	1	3
56	56	56	56	56	56	50	40	30	56
98	98	98	98	98	19	98	98	N/A	98

Table 7: Preliminary metrics derived from test results

	DT_1	DT_2	DT_3	DT_4	DT_5	DT_6	DT_7	DT_8	DT_9
pm_1	63.66	63.66	63.66	63.66	63.66	63.66	5.22	2.79	1.12
pm_2	76.34	76.34	76.34	76.34	76.34	76.34	7.38	1.69	0.74
pm_3	84.95	84.95	84.95	84.95	84.95	84.95	12.9	1.35	1.2
pm_4	75.91	75.91	75.91	75.91	75.91	75.91	16.1	3.99	2.8
pm_5	0	0	0	0	70	0	0	0	0
pm_6	0	0	0	0	70	0	0	0	0

pm_7	20	20	20	20	85	20	0	0	0
pm_8	65.68	65.68	65.68	65.68	65.68	65.68	55.65	47.63	35.4
pm_9	23	23	4	85.71	23	23	1.5	0.75	0.5
pm_{10}	1.9	92.435	1.9	1.9	1.9	1.9	1.54	1.6	1.33
pm_{11}	64	64	64	64	64	64	54.84	39.32	25.23
pm_{12}	0.31	0.31	0.31	0.31	0.31	80.52	0.31	0.31	0

Table 8: Overall metric values for each device type and objectivity combination

	Energy	Synch	Asynch	Critical	Secure	Burst
Av. PDA (DT_1)	23.05	26.22	27.85	16.54	45.63	44.87
High CPU util. PDA (DT_2)	40.20	39.53	36.90	26.07	55.69	60.21
Good throughput PDA (DT_3)	22.33	22.73	27.85	16.14	44.93	44.54
Poor throughput PDA (DT_4)	25.43	37.75	27.85	17.86	47.95	45.93
High network errors PDA (DT_5)	27.71	30.74	48.35	48.90	53.22	48.34
Low battery PDA (DT_6)	53.43	49.81	48.72	33.42	54.54	58.46
Av. laptop (DT_7)	17.01	13.90	15.20	8.79	13.85	17.56
Strong laptop (DT_8)	12.45	9.97	10.98	6.32	8.85	10.12
Powerful workstation (DT_9)	8.15	6.76	7.00	4.14	5.70	6.67

Table 9: Capability/incapability determination of six predefined device types for six predefined routing objectives

	O_1	O_2	O_3	O_4	O_5	O_6
DT_1	**Capable**	Incapable	**Capable**	**Capable**	Incapable	Incapable
DT_2	Incapable	Incapable	**Capable**	Incapable	Incapable	Incapable
DT_3	**Capable**	**Capable**	**Capable**	**Capable**	Incapable	Incapable
DT_4	**Capable**	Incapable	**Capable**	**Capable**	Incapable	Incapable
DT_5	**Capable**	Incapable	**Capable**	Incapable	Incapable	Incapable
DT_6	Incapable	Incapable	Incapable	Incapable	Incapable	Incapable
DT_7	**Capable**	**Capable**	**Capable**	**Capable**	**Capable**	**Capable**
DT_8	**Capable**	**Capable**	**Capable**	**Capable**	**Capable**	**Capable**
DT_9	**Capable**	**Capable**	**Capable**	**Capable**	**Capable**	**Capable**

Table 11: The threshold values mapped to the actual CPU utilisation (C), memory usage (M), and battery level (B)

	Energy efficient			**Synchronous**			**Asynchronous**			**Critical**			**Secure**			**Burst**		
	C	M	B	C	M	B	C	M	B	C	M	B	C	M	B	C	M	B
DT_1	48	82	49	-	-	-	95	95	85	44	81	46	-	-	-	-	-	-
DT_2	-	-	-	-	-	-	95	95	74	-	-	-	-	-	-	-	-	-
DT_3	50	86	49	29	68	36	95	95	85	47	89	48	-	-	-	-	-	-
DT_4	38	71	42	-	-	-	95	95	85	42	86	45	-	-	-	-	-	-
DT_5	26	64	34	-	-	-	31	60	37	-	-	-	-	-	-	-	-	-
DT_6	-	-	-	-	-	-	28	61	84	-	-	-	-	-	-	-	-	-
DT_7	70	95	59	75	95	61	95	95	85	95	95	69	23	57	40	28	59	45
DT_8	94	95	66	95	95	68	95	95	85	95	95	76	61	70	70	63	78	72
DT_9	95	95	N/A	95	95	N/A	95	95	N/A	95	95	N/A	82	78	N/A	77	81	N/A