IETF MANET Working Group
Internet-Draft
Intended status: Informational
Expires: December 2010

T. Ramrekha
E. Panaousis
C. Politis
WMN Research Group
Kingston University London

JUN 24, 2010

A Generic Cognitive Adaptive Module (CAM) for MANETs
draft-ramrekha-manet-cam-00.txt

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html

This Internet-Draft will expire on December 24, 2010.

Copyright Notice

Copyright (c) 2010 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document.
Please review these documents carefully, as they describe your rights and restrictions with respect to this document.

Abstract

This document describes a generic Cognitive Adaptive Module (CAM) that can be utilized in conjunction with Mobile Ad hoc Network (MANET) routing protocols. The main concept behind CAM is the fact that the provisioning of multimedia communications traditionally requires routing Quality of Service (QoS) guarantees [9]. Such a task is NP Complete in MANETs when QoS optimization is subject to more than one parameter [9]. Hence, the provisioning of soft QoS guarantees for effective and efficient routing in dynamic environments, as specified in [1], is the best alternative. However, the latter cannot be optimally achieved by using a single metric based path selection process or routing approach due to variations in both upper layer service QoS requirements and situational constraints. CAM provides interfaces to routing components such that protocols can be segmented into these components and can be easily made configurable and adaptive. For instance, the route selection process can be done in an adaptive manner to satisfy the requirements for effective and efficient routing. This is achieved by providing interfaces from the CAM core to various user defined components (e.g. Repositories Component) such that all components and component parts (e.g. DYMO reactive routing logic) can inter-communicate.

1. Introduction

The autonomous nature of Mobile Ad hoc Networks (MANETs) makes them suitable for deployment in various scenarios. In such scenarios, the routing QoS is defined by the service requirements but the achievable QoS is limited by network and scenario constraints. A detailed list of these requirements and constraints is presented in [1].

It can be deduced that rigid routing protocols based on a fixed route selection process that only consider single path metrics SHALL NOT perform optimally in such dynamically varying environments.

Although the network performance using such protocols MAY be satisfactory for specific scenarios, the routing approach performs sub-optimally for wider context usages.

An adaptive approach using routing logics from well tested protocols such as [3] and [4] will provide a more flexible routing solution for the widespread use of MANETs. CAM is a generic module that provides interfaces for user defined routing components e.g. "Routing algorithms", "Repositories" and "Route quality" determination so that these are easily configurable and reusable for different scenarios. In addition, CAM offers interfaces to "Monitor" and "Adaptive" components that allow protocols to cognitively adapt its routing process in dynamic environments. This SHOULD enhance overall efficiency and effectiveness (both terms defined as design requirements in [1]) of routing algorithms. This version of CAM defines the appropriate module, interfaces and components necessary to enable this. Furthermore, this document describes the operation of a modified version of [10] that uses CAM. This CML version utilizes [3], [4] and [5] in order to adapt to various network sizes and node distributions by utilizing CAM as a facilitator towards making the protocol cognitive and adaptive.

CAM optionally uses protocols and requirements defined in [3], [4], [5] and [10]. The module makes no assumption about the underlying link layer other than those made in [10].

2. Notation and Definitions

lg - The base 2 logarithm function, so that 2^lg(x) = x.

x||y - concatenation of bit string x and bit string y.

|x| - number of bits in string x.

3. Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [2].

3.1. CML Terminology

This section defines terminology associated with CAM that are not already defined in or that differs in meaning from the terminology

CAM Module - A collection of interfaces and generalized functions used to separate routing protocol components. Also defines components that are necessary for cognition and adaptation.

CAM Interface - A structure defining the elements required and provided by the module.

CAM Component - A structure containing definitions and implementations of routing logics and structures.

Routing algorithm - A logic that defines a process for data packet delivery including route selection.

Message passing - A method of transferring information from component parts to the CAM module and vice-versa.

Threshold - A threshold for the context being monitored in the network e.g. density of network, network size, battery levels of nodes, etc.

Upper_Threshold_C - The upper value of a threshold for a particular monitored MANET context such as network size, node density in the neighborhood and battery level of nodes.

Lower_Threshold_C - The lower value of threshold for a particular monitored context.

MANET Context - A set of characteristics describing the MANET and its environment as defined in [1]. This includes node mobility levels, small and large node groups as well as technological and environmental constraints such as limited battery life of devices and bandwidth limitations of wireless links.

Monitors - A part of the Monitor Component that defines and implements necessary logic for user defined monitoring requirements for MANET contexts.

Operation Band - The nodal routing state relative to the monitored Context being less than the threshold or more than threshold.

Coarse variables - This identifies the component that is being targeted. The value of the coarse variable MAY be equal to the ID of the component for convenience.
Fine variables - This identifies the parameter that needs to be modified within a component part. The variable value SHOULD uniquely identify variable parameters within each component part.

Trigger - An implemented logic in the CAM core that is used by the Monitors to activate specific Adaptive component parts when a threshold is exceeded.

4. Applicability

This module is designed to be used with protocols that follow recommendations particularly from [1] and [7]. It also extends the concept of protocol segmentation as introduced by [7] for reasons specified thereby. Most notably segments from protocols [3] and [4] can be used within such parts of the component for ease of implementation and configuration. A standard MANET routing protocol should be able to provide optimal QoS performances in all situations and the main goal of CAM is to facilitate such an optimization by adding cognitive and adaptive features to routing algorithms so that they are best fitted for all application purposes and easily configurable by users. Examples and particularities of some possible scenarios (also mentioned in [1]) where MANETs could be deployed are:

Emergency rescuer and military ad hoc communication - Rescuers and military participants will require multimedia communications (requiring low delay and delay jitter as well as high throughput routing QoS) in terrains where obstacles are common. Devices will have limited battery resources and the network topology (in terms of both network size and node distribution) will change regularly as participating nodes join or leave the network.

Mesh-based wireless community networks - Community users are likely to access multimedia services. Since the topology consists of static rechargeable routers, energy spent for routing is not a limitation towards QoS provisioning. However, users might prefer more efficient energy utilization for greener and cheaper solutions while also maintaining the required QoS routing levels.

Mesh-based wireless enterprise networks - Enterprise users (i.e. office users) are likely to access email and file transfer services (requiring low packet loss routing QoS). Since the nodes are rechargeable, energy limitation is not an issue but users might prefer more efficient energy utilization for greener and cheaper solutions.

Smart home ad hoc networks - Home users MAY want to distribute PCs. Here "bursty" communication would be desired and proactive maintenance of route information MAY be inefficient and expensive in idle periods between bursts.

In addition, CAM SHOULD be useful for communication devices with multithreading capabilities such as new generation PDAs. Moreover, CAM MAY also be used for general purpose MANETs.

5. Protocol Functioning and Overview

This document does not describe a protocol but a module that can incorporate parts routing protocols such as [3], [4] and [5]. Thus this section contains an overview of the module and its functional descriptions.

5.1. CAM Overview

This module is designed to work as a facilitator in the deployment of adaptive routing protocols for MANETs including adaptive hybrid approaches such as [10]. It is required that CAM MUST be installed in each MANET node. The module consists of the module core and components as shown in Figure 1. The core contains generic logic to enable data and control flow among components. Then, the pluggable components contain self-configurable logical parts that can be easily modified or defined by users.

```
   +------------------------------------------------------------------+
   | +-----+... +-----+   +-----+ ... +-----+        +-----+  +-----+ |
   | |PART |    |PART |   |PART |     |PART |        |PART |  |PART | |
   | |  1  |    | p   |   |  1  |     | p   |        | 1   |  | p   | |
   | +-----+    +-----+   +-----+     +-----+        +-----+  +-----+ |
   |  A  |        A  |      A  |        A  |          A  |     A  |   |
   |  |  v        |  v      |  v        |  v          |  v     |  v   |
   |+-----------------+   +---------------------+ . +---------------+ |
   ||    Component1   |   |    Component2       |...|  Component c  | |
   |+-----------------+   +---------------------+ . +---------------+ |
   |  A          |           A               |         A         |    |
   |  |          |           |               |         |         |    |
   |  |          v           |               v         |         v    |
   |+---------------------------------------------------------------+ |
   ||                           CAM CORE                            | |
   || Fine Variables                                                | |
   || Coarse Variables                                              | |
   || In/Out Component Interfaces                                   | |
   || In/Out Packet interfaces                                      | |
   |+---------------------------------------------------------------+ |
   +------------------------------------------------------------------+
```

Figure 1: Overview of the CAM core, components and parts

5.2. CAM Core

The module core contains essential data structures and logic definitions as well as implementations required by the module components. These SHOULD facilitate the configuration of well known routing parts into more adaptive and hybrid routing approaches.

5.2.1. Interfaces

The core MUST implement interfaces for each defined component. Thus, the core can communicate to the components through these interfaces. Component-to-Component communication MUST only be possible via the core. Each interface is associated with a component ID. The core can then pass messages to the required component using its stored ID within core data structures.

5.2.2. Component and part IDs

Each component and each component part is identified through a unique ID. This ID MAY be coded as a 'x' bit string where

x = x_component || x_part

|x_component| = (lg (Number of components))

|x_part| = (lg (Number of parts))

5.2.3. Thresholds and Triggers

The core MUST define triggers for all the contexts being monitored in the Monitor component parts. To achieve this, an upper and lower threshold MUST be defined for each monitored context. The trigger is used to contact the Adaptive module if for a context C:

1. (Previous_C_Value < Lower_Threshold_C) and (Current_C_Value>= Lower_Threshold_C) or

2. (Previous_C_Value > Upper_Threshold_C) and (Current_C_Value<= Upper_Threshold_C)

The trigger MUST contact the appropriate component by using the required IDs in order to change the operation band.

5.2.4. Component Coarse and Fine variables

The core MUST define both coarse and fine variables for component parts. This will allow the core to change routing behavior according to context changes. For each active component part, a coarse variable MAY be assigned the component part ID. Then, for each defined coarse variable specifying a part, associated parameter names and values are stored. These values are set by the Adaptive component parts and are used to change the routing behavior of protocols.

5.3. CAM Components

Each component consists of a group of logic parts that allows the utilized routing protocol to operate in a specific manner. While components SHOULD be user defined, some are essential. These are Monitor, Adaptive, Routing, Metric Specification, Packet and TLV specification and Repositories components as defined next.

5.3.1. Monitor

The Monitor component aims at providing cognition to routing protocols. The parts of this component should contain logic that processes incoming packets or data in the Repositories component or both, in order to derive network state information. These parts, also called monitors, checks the appropriate threshold according to the current operating band. It then alerts the CAM core if a threshold is exceeded. This SHOULD be done using the appropriate threshold trigger.

From the scenarios described above, important parts that MAY be defined in this component are monitors for:

Number of neighbors monitor - it collects and maintains information for number of neighbors of the node. This will give an indication of the density of the local network.

Rate of change in neighbors monitor - it regularly compares current neighborhood information to calculate neighborhood changes. Neighborhood changes include changes in the total number of neighbors and neighbor nodes leaving or joining the neighborhood. It then, computes and stores the value for the rate of change. This information is then stored in the Repositories component. This will give an indication of network mobility.

Total number of nodes monitor - this monitors the total number of nodes at regular intervals of time within the network. This function is specific to the routing algorithm being used. In proactive approaches such as [3], this function only consists of counting the number of rows in the routing table. In the case of [4], an estimation of the number of nodes in the network can be obtained by using probe packets such as in [10].

Traffic profile monitor - this monitor checks for the traffic profile of current data packets received at the node. It stores this profile along with the number of connections that are supported by this node and the number of packets received for each profile type within a timeout period. This SHOULD be stored in the Repositories component. This also specifies the metrics that SHOULD be recorded by the metric statistics monitor and determines the coefficient values for metrics used in the path selection process.

Metric Statistics - it processes the received routing control packets containing metric TLVs (similar to R_etx TLV in [OLSR-etx]) and stores the metric value of nodes in the Repositories component indexed to the appropriate node or path in a routing table. A few important metrics that can be stored in packet TLVs are estimated link error before successful transmission (ELTX), estimated link delay (ELD), estimated link bandwidth (ELBW), estimated link delay jitter (ELJ), and neighbor node energy level (NEN). The values for route quality can be calculated as follows:

- ETX: The ELTX is the estimated number of transmissions required to successfully send a packet over a link as defined in [11] and [13]. ETX is defined as the sum of the ELTX values of links that form a given path.

- ED: It is assumed that the clocks in all the participating nodes are synchronized. The ELD value can then be calculated using a timestamp message TLV that is written by each sender. The receiver node on the end of the link then has to use the current clock value. The difference between the two clock values gives the ELD value. Since delay is an additive metric, the ED value of a path is equal to the sum of all ELD values of links within that path.

- EBW: The ELBW value MAY be calculated using the ELD value of a link as estimated above.

  ELBW = Received Packet size/Link ELD

Since bandwidth of a path is constrained by the minimum ELBW along the path, the EBW value is equal to the minimum ELBW value. Another alternative for calculating EBW is described in [9].

- EJ: The EJ value is additive in nature. It is the sum of ELJ values where each ELJ value is the variance in consecutive ELD values for that link.

- NEN: The NEN value can be included by each node in a TLV when it sends control packet messages to neighbor nodes. If the energy level of a route is required, the sum of NEN can be sent in node TLVs that are incremented at each node along that route.

- Hop Count (HC): The HC value can be obtained from the message header <msg-hop-count> field defined in [7].

5.3.2. Adaptive

The Adapt component MUST check the validity of a trigger. Furthermore, they MUST change coarse and fine variables in the CAM core so that adaptive actions are enabled. The parts of this component should contain implementations of actions that change the routing behavior of nodes. Therefore, the triggers should target one or more adaptive parts as required by the utilized adaptive concept. Hence, the parts here SHOULD allow users to specify their desired adaptive actions.

As mentioned above, the required module part MUST first check if the trigger is valid by confirming whether the threshold has been exceeded. This is done by consulting the Repositories module (for a given time period) or by initiating a confirmation process. Then, if the trigger is valid, some Adaptive parts and their possible roles MAY be:

- Switching routing logic - the adaptive module MAY decide to switch from proactive routing to reactive routing when triggers such as the one for the number of nodes or number of neighbors thresholds are exceeded.
- Tuning route discovery and maintenance intervals - the logic in such a part MAY change the interval for route discovery and maintenance such as HELLO intervals and route timeouts. This MAY be as a result of a confirmed mobility threshold trigger.
- Determining coefficient values for routing metrics - the coefficient values established here determines the importance

of each routing metric as a result of traffic requirements and scenario specific constraints (user MAY input these manually).

- Other self defined parts - other similar logics can be used in parts to define actions required to adapt to changes in context as detected by the monitors and alerted through confirmed triggers.

5.3.3. Routing algorithms

For the purposes of CAM, this component MUST have at least one part defined whereby all the essential routing heuristics necessary for routing packets in MANETs are defined. The routing component contains algorithms as parts of protocols defined in [3], [4] and [5]. Hybrid approaches MAY therefore be enabled by defining a hybrid routing logic within a single part. Otherwise routing logics of different approaches MAY be defined several parts. These separate parts MAY be then triggered when deemed necessary. The processes of route discovery, route maintenance and route selection should be defined within at least one part.

5.3.4. Route quality determination

This component is used to define the logic for quantifying route qualities using defined metrics. Here, the parts define metrics that are required by the "Metric Statistics" part of the "Monitor component". This SHOULD allow for a multiple metric based path selection process. Several techniques MAY be utilized for metric quantization. Firstly, hierarchical based metric quantization MAY be implemented where route selection is based on comparing high priority metrics of routes (that SHOULD be above a certain defined quality level) followed by comparison of lower quality route metric values as required by supported services:

```
if (ETX > ETX_min_quality)

 if (Metric2 > m2_min_quality)

 . . .

 if (Metric(m-1) >m(m-1)_min_quality)

set Route_quality = Value_m
```

Moreover, a utility score for each route Rt MAY be used to make route selections [8] based on the metrics that are defined:

U(Rt) = (a*ETX) + (b*ED) + (c*EBW) + (d*EJ) + (e*NEN) + (f*HC).

Then, a hybrid approach combining both hierarchical and utility score techniques can be used for route quality determination.

The route quality information MUST be stored in the repositories component and indexed to the appropriate routes.

5.3.5. Packet and TLV Specification

A generalized packet format MUST be used so that all parts and components can access information within packets. This module is described in section 7.

5.3.6. Repositories

This component is used to store data that are useful for other components of the CAM module. It is further described in section 8.

5.3.7. Security component

The security component contains logic that enables security measures against attacks launched by malicious nodes in the network. A detailed description of this component is presented in section 12.

5.3.8. User defined Components and parts

The CAM MUST be able to accommodate user defined components and parts.

6. Protocol Operation

6.1. Control and Data flows

The control and data flows are controlled by the CAM core. An overview of its stepwise operation is described next.

If a control packet is received from another node:

1. The packet is duplicated. One copy is used to extract context information in the monitor component while the other is used by the routing algorithm for processing route information.
2. The routing component is identified using the coarse variable that identifies the "Routing algorithms" component.
3. Any associated fine variable values SHOULD be used to identify parts and to update the routing parameters such as that of route discovery intervals.

4. The monitor gathers information in the Repositories component. It also compares the network context against defined thresholds according to section 5.2.3.
5. If a threshold is exceeded:
    a. The associated trigger is used to pass the message to the relevant Adaptive component part using its associated ID.
    b. The Adaptive component:
        i. MUST check whether the trigger is valid.
        ii. MAY change values for coarse and fine variables if trigger is valid.

If a control packet needs to be sent to another node:

1. The routing algorithm sends the packet to the core packet send interface.
2. The core sends the packet to the outgoing queue.

If a data packet is received from another node:

1. The core sends the data to the Routing algorithm component.
2. The Routing algorithm component checks the packets according to the routing logic.
    a. If the current node was the intended recipient of the packet:
        i. Data packet is sent to the core send interface.
        ii. Data packet is placed in the incoming queue and then sent to upper OSI layers.
    b. Else, the data packet is forwarded as specified by the routing logic.

If a data packet needs to be sent to another node:

1. The core sends the packet to the relevant part of the "Routing algorithms" component.
2. The data packet is forwarded as specified by the routing logic.

6.2. CAM use case: ChaMeLeon (CML)

This section describes the way CAM can be utilized to implement a hybrid adaptive routing protocol such as a modified version of [10]. This version of CML SHOULD provide more flexibility as compared to OLSRv2 and DYMO individually. Consequently, it MAY be more

battery limitations and packet delay are important routing factors. In such a context, this version of CML MAY provide better performances. CAM allows for easier protocol configuration and thus add-ons can be readily integrated to proposed protocols while routing parameters MAY be manually or automatically tuned.

The Routing Algorithms component of CML consists of various parts that include a proactive part based on OLSRv2, a reactive DYMO part and a neighbor discovery part [5]. Initially, routing is carried out as described in [3]. The Monitor component monitors the network size and network density. The [5] part is always operational updating neighborhood information as described in [3].

The coarse and fine variables indicate the mode of operation and the default mode of operation is [3]. The node operations can be described as follows:

1. The monitor component:
    a. Monitors network size by counting the number of nodes in the network in the routing table.
    b. Monitors the network density by counting number of neighbors in the 2-hop neighborhood.
    c. Compares these values with the corresponding threshold values that MAY be stored in the core or in repositories. These threshold values MAY be lower or upper bound threshold values for density and network size depending on the operation band of the node.
    d. Uses the corresponding trigger as defined in the core to access the relevant part of the adaptive module.
2. The adaptive component:
    a. Uses the core to set the TC_HOP_LIMIT value to H1 hops if the fraction of (network density/Total number of nodes) is greater than ratio value R.
    b. Else uses the core to set the TC_HOP_LIMIT value to H2 hops if the value of node density is greater than D1 and number of nodes greater than N.
    c. Else uses the core to set the TC_HOP_LIMIT value to 2 hops if the value of node density is less than D1 and number of nodes greater than N.
    d. Uses the core to increment the TC_HOP_LIMIT value by 1 if the value of node density is greater than D2 and number of nodes is less than N.

3. The Routing Algorithms component:
    a. Consists of [3], [4] and [5] parts.
    b. Uses routing logic as described in [3] by default to:
        i. Calculate routes and store them in routing tables proactively using HELLO and TC messages.

ii. Forward data to destinations found in the proactive routing tables.

iii. Process control packets defined in [3] as specified in [3].

iv. Process packets defined in [4] as follows:

1. RREQ packets are unicasted to destinations found in the proactive routing table.
2. RREQ are flooded through MPR nodes if destination is not in proactive routing table and relevant information stored in the reactive routing table as specified in [4].
3. RREP are unicasted towards the source node using the reactive table RREQ information if the proactive table does not contain an entry for such a source.
4. If the proactive table has such an entry, it sends the packet through that route updating the reactive table with relevant entries.

c. Uses routing logic as described in [4] to:

i. Generate reactive routing packets when routes to destinations are not found in the proactive table.

ii. Forward data to such destinations not found in the proactive routing tables but listed in the reactive table as a result of the previous step.

4. The Repositories component:

a. Defines and implements the reactive, proactive, neighborhood routing tables as well as other tables as required in [3] and [4].

b. Defines and implements data structures to store values required by the other specified components.

5. The Packet and TLV specification component defines and produces packets and messages using the formats specified in [3], [4], [5] and [7] respectively.

```
   +-----------------------------------------------------------------+
   | +-----+ +-----+   +------+ +-------+     +------+ +----+ +----+ |
   | |Net. | |Net. |   |TC Hop| |Routing|     |OLSRv2| |DYMO| |NHDP| |
   | |Size | |Den. |   |Limit | | Part  |     |      | |    | |    | |
   | +-----+ +-----+   +------+ +-------+     +------+ +----+ +----+ |
   |  A  |     A  |      A  |      A  |          A  |    A |    A |  |
   |  |  v     |  v      |  v      |  v          |  v    | v    | v  |
   |+--------------+   +-----------------+   +----------------------+|
   ||    Monitor   |   |     Adaptive    |   |  Routing Algorithms  ||
   |+--------------+   +-----------------+   +----------------------+|
   |  A          |            A     |              A         |       |
   |  |          |            |     |              |         |       |
   |  |          v            |     v              |         v       |
   |+--------------------------------------------------------------+ |
   ||                            CAM CORE                          | |
   || Fine Variables                                               | |
   || Coarse Variables                                             | |
   || In/Out Component Interfaces                                  | |
   || In/Out Packet interfaces                                     | |
   |+--------------------------------------------------------------+ |
   +-----------------------------------------------------------------+
```

Figure 2: Overview of the updated CML protocol with CAM

7. Packet and Message Formats

The allowed Packet and Message formats are specified in [7] and hence all packets as well as messages are formatted with the generalized packet format. All packet types required by the "Routing Algorithms" component MUST be defined. Furthermore, TLVs for each packet type or message type MUST be defined according to the requirements of the other components.

8. Repositories

This component contains the necessary data required for the operation of other components. The parts contain logic that defines and implements data structures such as tables to store required component information. The module MUST contain at least one part that defines the necessary data structures and the processes that allow storage of data as required by the other components. This data MUST be retrievable by all components via the CAM core.

9. Constants

9.1. Network Threshold and other parameter Values

The considered constants Upper_Threshold_C, and Lower_Threshold_C values as well as coefficients a, b, c, d, e and f are found through experimentation or research results. These are dependent on the module components that are defined by the users and MUST be modifiable by the "Adaptive" Component.

In the case of CML the values for H1, H2, R, D1, D2, and N are also determined using experimental and research results. Other parameter values are specified in [3], [4] and [5].

10. Message Emission and Jitter

Synchronization of control messages SHOULD be avoided as mentioned in [6].

11. IPv6 Considerations

All the operations and parameters described in this document can be used for both IP version 4 and IP version 6. For IPv6 networks, the IPv4 addresses in the utilized packets and messages need to be replaced by IPv6 addresses. The packet and message sizes will also increase accordingly.

12. Security Considerations

CAM SHOULD have a security component where logics for security measures are defined. There are special security considerations for MANET routing protocols as described in [16]. Initially, authentication, confidentiality and integrity MUST be satisfied for any MANET communication path. In this way, legitimate nodes are protected against external malicious entities. On the other hand, intrusion detection mechanisms MUST be designed to defend MANET nodes against compromised entities, node capture attacks or jammers. There are more critical security vulnerabilities since encryption, digital signatures and hashing do not satisfy them. In other words, smart detection and prevention mechanisms MUST be adopted by CAM to recognize any adversary that has succeeded to penetrate the first wall of defense. It is worth noting here that MANET routing protocols based on game theoretic concepts such as the one published in [17] might be adopted in CAM to provide energy efficient and more precise intrusion detection.

Malicious entities might launch attacks against one or more components of CAM thus a security module MUST provide protection, privacy and trustworthiness focusing on the following CAM functionalities:

- Monitor: adversaries might corrupt the nodes' monitoring process by launching attack such as jamming or pretending other legitimate nodes to attract traffic and start dropping packets on a MANET path.
- Adaptive: adversaries might advertise that a threshold has been reached and the routing behavior must be changed accordingly although this is not true.
- Routing algorithm: adversaries may corrupt the functionality of the routing algorithm selection. In this case MANET nodes do not choose the appropriate and efficient routing protocol thus the performance metrics might significantly decrease.
- Route quality determination: adversaries might advertise different requirements than legitimate nodes and thus wrong decisions about the network's QoS preferences are taken.

13. IANA Considerations

The values assigned to messages and message types must in conformance with the ones defined in specifications within [3], [4], [5] and [12].

14. Conclusions

This I-D presents CAM, a module that aims at easing the configuration of MANET routing protocols. This is an extension of the concept of segmenting [5] from [3]. In this document, the objective is to encourage designs for more flexible routing protocols by segmenting the routing logic into segregated components that can be easily configured and rendered adaptive. Thus, such a routing protocol will be better suited for a wider range of real life applications. A modified version of [10] that integrates [3], [4] and [5] routing logics is also presented to demonstrate this. This version of CML MAY be better suited than its individual component parts if they were all to be used for a given scenario e.g. disaster multimedia communication where data delivery latency has to be minimized, the battery power of devices is limited while the network size as well as node distribution are bound to change. Thus CAM SHOULD allow for easier deployments of MANETs in a wider range of applications.

15. References

15.1. Normative References

[1] Macker, J. and S. Corson, "Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations", RFC 2501, January 1999.

[2] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[3] Clausen, T., Dearlove, C., and P. Jacquet, "The Optimized Link State Routing Protocol version 2", draft-ietf-manet-olsrv2-11.txt (work in progress), March 2010.

[4] Chakeres, I., and Perkins, C., " Dynamic MANET On-demand (DYMO) Routing", draft-ietf-manet-dymo-19.txt (work in progress), March 2010.

[5] Clausen, T., C. Dearlove, and Dean, J., "MANET Neighborhood Discovery Protocol (NHDP)", draft-ietf-manet-nhdp-12.txt (work in progress), March 2010.

[6] Clausen, T., Dearlove, C., and B. Adamson, "Jitter considerations in MANETs", RFC 5148, February 2008.

[7] Clausen, T., Dean, J., Dearlove, C., and Adjih, C. "Generalized MANET Packet/Message Format", RFC 5444, February 2009.

[8] Dearlove, C., Clausen, T., and P. Jacquet, "Link Metrics for OLSRv2" (work in progress).

15.2. Informative References

[9] L. Chen and W. Heinzelman, "Qos-aware routing based on bandwidth estimation for mobile ad hoc networks," Selected Areas in Communications, IEEE Journal on, vol. 23, no. 3, pp. 561-572, March 2005.

[10] Ramrekha, T., Panaousis, E., Millar, G. and Politis, C., "ChaMeLeon (CML): A hybrid and adaptive routing protocol for Emergency Situations" draft-ramrekha-manet-cml-00.txt (work in progress), February 2010.

[11] De Couto, D., Aguayo, D., Bicket, J., and R. Morris, "A High-Throughput Path Metric for Multi-Hop Wireless Routing", Proceedings of the MOBICOM Conference, 2003.

[12] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", RFC 5226, BCP 26, May 2008.

[13] Rogge, H., Baccelli, E., and Kaplan, A.," Packet Sequence Number based ETX Metric for Mobile Ad Hoc Networks" draft-funkfeuer-manet-olsrv2-etx-01 (work in progress), March 2010.

[14] Chakeres, I., "IANA Allocations for MANET Protocols", RFC 5498, March 2009.

[15] Clausen, T. and C. Dearlove, "Representing multi-value time in MANETs", RFC 5497, March 2009.

[16] Anjum, F. and Mouchtaris, P. "Security for Wireless Ad Hoc Networks", ISBN: 978-0-471-75688-0, John Wiley & Sons, March 2007.

[17] Panaousis, E.A and Politis, C. "A game theoretic approach for securing AODV in emergency Mobile Ad Hoc Networks," Local Computer Networks, 2009. LCN 2009. IEEE 34th Conference on, vol., no., pp.985-992, 20-23 Oct. 2009, doi: 10.1109/LCN.2009.5355020

16. Acknowledgments

The authors wish to acknowledge the support of the ICT European 7th Framework Program and all the partners in PEACE (IP-based Emergency Applications and services for next generation networks) project with contract number 225654. In particular, the input from Mr. Grant P. Millar, Researcher at WMN Group Kingston University, was well appreciated.

This document was prepared using 2-Word-v2.0.template.dot.

Authors' Addresses

The following researchers who have contributed to this I-D are members of the Wireless Multimedia and Networking (WMN) Research Group at Kingston University London:

Tipu Arvind Ramrekha
Researcher, WMN Research Group
Kingston University London KT1 2EE

Phone: (+44) 02084177025
Email: a.ramrekha@kingston.ac.uk

Emmanouil A. Panaousis
Researcher, WMN Research Group
Kingston University London KT1 2EE

Phone: (+44) 02084177025
Email: e.panaousis@kingston.ac.uk

Christos Politis
Head of WMN Research Group
Kingston University London KT1 2EE

Phone: (+44) 02084172653
Email: c.politis@kingston.ac.uk