A Content-Centric Architecture for Green Networking in IEEE 802.11 MANETs

Marica Amadeo, Antonella Molinaro, and Giuseppe Ruggeri

University "Mediterranea" of Reggio Calabria - DIMET Department
Loc. Feo di Vito, 89100 Reggio Calabria, Italy
{marica.amadeo,antonella.molinaro,giuseppe.ruggeri}@unirc.it

Abstract. In this paper we aim to demonstrate that the emerging paradigm of content-centric networking conceived for future Internet architectures can be also beneficial from the energy efficiency point of view. The reference scenario to prove this statement is a Mobile Ad hoc Network (MANET) characterized by dynamic topology and intermittent connectivity. We design CHANET, a content-centric MANET that relies on a connectionless layer built on top of legacy IEEE 802.11 networks to provide energy-efficient content-based transport functionality without relying on the TCP/IP protocol suite.

Keywords: Content Centric Networking, Green Networking, MANETs.

1 Introduction

Mobile ad-hoc networks (MANETs) are self-organized networks of battery-powered devices that exchange information without relying on any centralized control or pre-existing network infrastructure. IEEE 802.11 [1] MANETs represent today a pervasive low-cost wireless technology thanks to the widespread diffusion of diversified 802.11-enabled handheld devices (like smartphones, tablets, MP3 players). MANET devices can actively cooperate to forward data over multihop paths towards a destination node, which can be either *any* node in the MANET or a *gateway* node offering connectivity to the Internet.

The primary usage of the current Internet as a means for discovering, uploading, accessing and sharing contents, is asking for a radical change in the underlying communication paradigm, from an *address-centric* to a *content-centric* model [2]. Several research projects are based on this idea and suggest a clean slate architecture design to build the future Internet [3] [4].

The content-centric, or information-centric, vision enables the network to focus on *what* instead of *where* data can be retrieved, through *naming* data contents instead of their location (*IP addresses*). This approach allows separating trust in data content from trust in data paths (i.e., transmission channels, hosts and servers) by naming the data through security mechanisms, with the additional advantage to enable *in-network* data caching/storing to optimize traffic management. In a content-centric network, communication is driven by receivers, which ask for a given content typically by broadcasting an *Interest* packet.

Joel J.P.C. Rodrigues et al.: (Eds.): GreeNets 2011, LNICST 51, pp. 73–87, 2012.

The network can satisfy the request by forwarding it to any node holding a copy of the requested content.

This decoupling in space and time between senders and receivers make content-centric networking an appealing solution also for environments with intermittent connectivity like MANETs. Recent works in the literature have shown potential beneficial effects of the content-centric paradigm in MANETs [5], [6], [7]. In this paper, we aim at designing a feasible architecture for supporting content-centric communications in a MANET; we intend to investigate whether this approach can be also beneficial from the energy efficiency point of view, and the extent of this benefit.

The proposed content-centric architecture is called CHANET (Content centric fasHion mANET). It is based on a *connectionless* content-centric layer designed on top of the IEEE 802.11 Data Link layer [1], which exploits only *broadcast* packets and *named* contents by letting each receiving node take *local* forwarding decisions based on packets overhearing.

CHANET is especially conceived to cope with dynamic topologies and intermittent connectivity with the aim of:

- keeping signalling overhead and power consumption very low;
- leveraging simplicity, availability and robustness of packet broadcasting and overhearing while keeping scalability (and broadcast storm [8]) under control by means of in-network caching and smart dissemination techniques;
- reducing power consumption by also relying on *local* forwarding decisions without explicit signalling exchange with neighbours;
- inherently supporting mobility of source and destination nodes in the MANET;
- indirectly implementing functions like error control and retransmissions, traditionally implemented at transport layer;
- providing benefits to all involved parties: to users (by allowing fast and low-power access to the requested content), and to content and network providers (by reducing operational costs of the provided sources and infrastructure).

The remainder of the paper is organized as follows. Section 2 briefly summarises related work in the area of content-centric MANETs; Section 3 describes the proposed CHANET architecture; simulation results are reported in Section 4 that show the CHANET performance against the traditional address-centric Internet model. Conclusive remarks are reported in Section 5.

2 Related Work

Content-centric network architectures proposed for future Internet [3], [4] share some common functions:

- *Content naming and security* – contents are provided with globally unique names from a flat or hierarchical space. Names are often self-certified to securely verify the authenticity of the content and the publisher.

- *Content discovery* – network nodes cooperate to forward content requests towards one or more nodes that store the content.
- *Content delivery* – it consists in forwarding the discovered content from a storage node to the subscriber. The mechanisms can exploit either route information encoded in the packets header or they keep track of each forwarded request by caching information about the interface where the request has been received from. This interface will be used to subsequently forward the retrieved data over the path back to the subscriber, like in [2].
- *In-network content caching* – network nodes can temporarily store forwarded contents, so that they can directly send the data back to the requesting node, instead of forwarding the request upstream.

Authors of [5] and [6] showed the effectiveness of a content-centric approach in a MANET. They designed a topology-agnostic data-centric forwarding protocol, named Listen First, Broadcast Later (LFBL), that exploits packet overhearing by each intermediate node to limit the drawbacks of broadcasting interests. All forwarding decisions are taken by the receiver with a minimal amount of state in each node. The data exchange phase follows the rules of *distance-based* forwarding with collision avoidance. Performance evaluation shows that LFBL significantly outperforms the traditional Ad hoc On-Demand Distance Vector (AODV) protocol [9] in highly dynamic environments.

In [7], the content-centric approach is implemented on a large scale tactical/emergency MANET with high mobility and lossy channels. Representative experiments show the superiority of the proposal over traditional proactive routing like Optimized Link State Routing Protocol (OLSR).

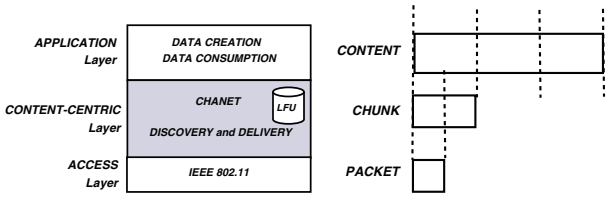
Energy efficiency is a critical requirement in MANETs due to the battery-powered nature of mobile device. Considerable research has been devoted in the past to low-power protocol design in an effort to enhance energy efficiency of MANETs [10]. On the other hand, in [11] and [12], the authors proved that the content-centric networking of [2] opens new possibilities for energy-efficient content dissemination in wired scenarios compared to traditional content delivery networks, where content is fetched from the origin server. Energy saving mainly comes from reducing hop counts by storing content at the intermediate nodes. Content-centric communications revealed to be energy-efficient also in sensor network environments [13] where used to manage short packets in a stationary network.

In this paper, we are interested to assess effectiveness of the content-centric approach in the dynamic MANET environment, where nodes are solicited to exchange a significant amount of data between mobile source and receivers.

3 The CHANET Architecture

As shown in Fig. 1(a), the CHANET architecture relies on a content-centric *connectionless* layer built on top of the 802.11 Data Link layer.

CHANET defines three message types for content discovery and delivery, illustrated in Fig. 2: *Interest* used to request the first content chunk, *Int-Ack* used



(a) Protocol Stack (b) Packetization process

Fig. 1. CHANET protocol architecture

to request subsequent chunks and to acknowledge previously received packets, and *Data-Object (D-Object)* containing data chunks.

The CHANET communication model is based on two phases: *(i) content discovery*, in which the consumer sends the first *Interest* to search for a given content; and *(ii) content delivery*, in which *D-Object(s)* are transferred to the intended receiver while new chunk requests and acknowledgements for previous chunk(s) are sent by means of *Int-Ack* packets.

The proposed architecture is *independent* on a specific naming scheme used at the application layer, provided that the name of the searched content is passed from application to the CHANET layer. In the conceived framework, each data content (e.g., a MP3 file, a YouTube video, proximity advertising information) has its own unique and persistent name, called *Content Identificator* (*CID*). Any content is divided into *chunks*. Each chunk has its own identifier (*chunkID*) and is provided with additional control information, e.g., the digital signature for securing it. The CHANET layer implements chunk *fragmentation and reassembly*: chunks are fragmented into packets before passing through the link layer (Fig. 1(b)). The size of chunks and packets is not fixed, but it can be decided by CHANET taking into account information from lower layers (e.g., bit error rate, channel quality).

Contents sources in the MANET can be either fixed stations (e.g., an access point playing the role of a gateway to the Internet) or mobile devices. The content can be originally owned by the node (e.g., photos locally uploaded from a camera) or downloaded by a remote server and available for distribution in the MANET. Sources are allowed to periodically send *Content Advertisement* messages in the MANET to spread information about the owned or downloaded content. Fixed nodes, like access points (APs) can exploit the periodic beacon transmissions to piggyback their content advertisements, while mobile nodes can deactivate such a feature in order to save energy.

Each CHANET node maintains a *Content Store (CoS)* to cache temporarily contents, thus becoming itself a source. Usually, such nodes do not advertise the stored content and do not forward any received advertisement for energy conserving purposes. They only may send data in response to a content request. To save energy, CHANET nodes do not cache all overheard contents, but only those matching a pending *Interest*. Furthermore, due to limited memory, cache must be periodically purged: CHANET deletes from cache *the least frequently used*

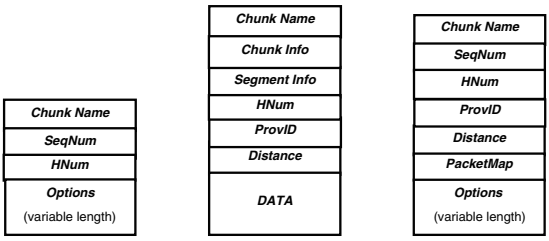
content. Implementation of more sophisticated policies are planned for future work. In the following, we refer to any node requesting a content as a *consumer*, and to any node that may satisfy the request (either the origin source or a node that keeps the content in its cache) as a *provider*.

In analogy to [2], each CHANET node maintains a *Pending Request Table* (PRT) for pending *Interest* and *Int-Ack* packets. In addition, also a *Content Provider Table* (CPT) is maintained by CHANET nodes, to keep the *nodeIDs* of the discovered providers and the distance to them. Nodes do not need an IP address: CHANET relies on MAC addresses as unique node identifiers. We recall that, since all communication is broadcast, 802.11 protocol cannot use retransmission mechanisms in case of packet collisions or losses. All retransmissions have to be coordinated by the CHANET layer.

3.1 Content Discovery

The content discovery phase relies on a *counter-based* broadcasting scheme for *Interest* forwarding. Counter-based schemes inhibit a node from broadcasting a packet, based on the number of packet copies already received by the node within a random access delay time. This technique is useful to reduce redundancy (and power consumption) and cope with the broadcast storm [8].

To request the first content chunk, a consumer C broadcasts an *Interest* that includes the "Chunk Name" in the form *CID/chunkID*, and waits for an answer. Answer can be either the reception of the requested *D-Object* sent by a provider, or the overhearing of the same *Interest* forwarded by a neighbouring node. If the consumer C does not detect any answer, CHANET schedules a new *Interest* transmission after a random defer time to reduce collision probability. If a *D-Object* is not received after a given number of attempts, CHANET declares the content *unreachable*.



(a) Interest (b) D-Object (c) Int-Ack

Fig. 2. CHANET packet types

As shown in Fig. 2(a), the CHANET *Interest* includes: the *Chunk Name*, a *Sequence Number* (*SeqNum*) to prevent message duplication, and a *Hop Number* (*HNum*) field that contains the number of hops the packet has crossed. *HNum* is increased by each node forwarding the packet: when it reaches its maximum

value (*MaxHops*), the *Interest* is discarded. Optional fields related to the naming system can be added in order to better qualify the content that matches the *Interest* (e.g., the Publisher Public Key Digest), or to limit the area where the reply might come from (e.g., the Scope).

On the *Interest* reception, each node applies the Processing Algorithm 1.

Algorithm 1. *Interest* Processing

```
1: if ((HNum == MaxHops) or (SeqNum is duplicated)) then
2:    Discard the Interest
3: else if (A matching is found in the CoS) then
4:    Compute the D-Object Defer Time d_d
5:    Wait(d_d)
6:    Send the D-Object
7:    Discard the Interest
8: else if (A matching is found in the PRT) then
9:    Discard the Interest
10: else
11:    Compute the Interest Defer Time d_i
12:    while (d_i is not elapsed) do
13:       Listen to the channel
14:       if (The Interest or the D-Object is detected) then
15:          Discard the Interest
16:          return
17:    Broadcast the Interest
18:    Insert the Interest in the PRT
19: return
```

More specifically, if the *HNum* field has not reached the *MaxHops* values and there is no duplication, a receiving node tries to find a match with cached content in its CoS. If a matching is found, then the node behaves like a provider P; it computes a *random defer time* d_d and waits before transmitting the *D-Object*. In case of CoS failure, it tries to find a match in its PRT. If a matching is found, then the *Interest* is discarded since a request for the same content has just been sent; otherwise, the node schedules the *Interest* re-broadcast after a *random defer time* d_i. CHANET assumes $d_d < d_i$ in order to give higher access priority to *D-Object* over *Interest* transmission.

Nonetheless the counter-based approach, some duplicated *Interests* could reach a provider P due to hidden terminal phenomena. CHANET assumes that P only replies to the first request and discards the others.

3.2 Content Delivery

The content delivery phase starts upon reception of an *Interest* by a provider P storing the requested content. The provider sends the stored *D-Object* after the *random defer time* d_d to avoid collisions with other nodes storing the data.

As shown in Fig. 2(b), a *D-Object* includes the following fields: *Chunk Name*, which is the same as for the *Interest*; *Chunk Info*, which contains security and temporal information related to the transferred chunk; *Segment Info*, used for chunk reassembly at the consumer side, since the chunk may be fragmented in more *D-Object(s)* before passing to the link layer; *ProvID*, which contains the provider's *nodeID*; *HNum*, which contains the number of hops the packet crosses (as for the *Interest*); *Distance*, which contains the hop distance between P and C, obtained from the *Interest*'s *HNum* field.

On *D-Object* reception, any intermediate node that maintains the related *Interest* in its PRT has to cancel it, to cache the data in its CoS, and to insert the newly discovered provider identifier (*ProvID*) in the CPT, including the distance to it, and the owned Content Name (identified by the *CID/ChunkID*). If the CPT entry already exists, the intermediate node only updates it. CPT entries are refreshed after any *D-Object* reception, otherwise they are purged when a *timeout* expires. Finally, the node rebroadcasts the *D-Object* after a random defer time and by using the *counter-based* approach, in analogy to the *Interest* forwarding. Nodes without a related PRT entry simply discard the packet.

By following the chain of PRT entries, the *D-Object(s)* is (are) forwarded to consumer C. If more providers have returned the data, consumer C inserts the providers' *nodeIDs* in its *CPT*; it selects the provider that may give the best performance and sets it as "preferred provider" (*PProv*). At this time, the selection algorithm depends only on the hop-count metric, thus C selects the *nearest* provider. More sophisticated choices could be implemented in the future.

After the first chunk reception and the provider selection, successive chunk requests will be broadcasted in *Int-Ack* packets, whose format is shown in Fig. 2(c). Compared to the *Interest*, the *Int-Ack* packet has three more fields: *ProvID* is the identifier of the *PProv* selected by consumer C; *Distance* represents the expected hop number between C and *PProv* (as read by the *HNum* field of previously received *D-Object*); *PacketMap* is used by C to acknowledge packets of the previously received chunk(s) so that the corrupted or lost ones may be retransmitted. *PacketMap* is a matrix whose generic element p_{ij} represents packet j (with j ranging from 1 to the number of packets in a chunk, p_{num}) in chunk i (with i ranging from 1 to a given number of recently received chunks, c_{num}). A value *1* in bit p_{ij} indicates that packet j in chunk i has been received correctly.

The *Int-Ack* may therefore carries two different content requests: the request for a new content chunk and the request for *D-Object(s)* associated to previous chunk(s) that has (have) to be retransmitted. It may happen that a receiving node has in the CoS the *D-Object(s)* to be retransmitted but not the new content chunk. In this case, we refer to *Partial CoS Matching*, while a *Total CoS Matching* happens when the node can satisfy both requests.

As for the *Interest*, *Int-Ack* processing is based on the *defer time* calculation and the *counter-based* forwarding, but three new features are introduced to improve performance in many aspects, including energy efficiency and scalability: *Int-Ack Aggregation*, *Provider Handoff* and *Selective Response*. Specifically, at

any forwarding node F, if the *Int-Ack* packet is still valid, a *Total CoS Matching* is first searched. If the content matches and the node is also the *PProv*, it checks if the same request has been just satisfied for other users. This check, referred as *Int-Ack Aggregation*, allows F to discard redundant requests from neighbouring nodes asking for the same content. If the *Int-Ack Aggregation* check fails, then F immediately sends the *D-Object*. Otherwise, if the forwarding node is not the *PProv*, it considers its distance to consumer C (by reading the *HNum* field in the *Int-Ack* packet) and compares it with the *Distance* field value. If F is closer to C than *PProv*, then it schedules the transmission of the *D-Object* after a defer time d_d. During the waiting time, a counter-based algorithm is run by overhearing *D-Object* packets that could be sent. This strategy, which we call *Provider Handoff*, mainly help to cope with highly dynamic topologies: due to the node mobility, a new provider can join the MANET and offer a better service than the current preferred provider; conversely, the consumer C can move away from the current *PProv* and enter the transmission range of a new provider.

If the *Total CoS Matching* fails but a *Partial CoS Matching* exists and F is closer to C than *PProv*, F may apply the *Selective Response* routine. It consists of sending the *D-Object(s)* that the *PacketMap* requires to retransmit and then forwarding a modified *Int-Ack*, where the *PacketMap* is purged of the just transmitted packets. As in the previous cases, a counter-based algorithm is run before transmission.

If there is a matching neither in the CoS nor in the PRT, F checks whether it is on the path between C and *PProv* by looking in its CPT. If a matching is found, a counter-based algorithm decides if forwarding the packet or not. The complete *Int-Ack* processing procedure is shown in Algorithm 2.

4 Performance Evaluation

The CHANET architecture has been implemented in Network Simulator 2 (ns-2) [14] for performance evaluation. The reference scenario is illustrated in Fig. 3. We consider one origin content source fixed in the centre of a square grid of side $500m$. The source can be co-located in an AP, which represents the only infrastructure made available by the network operator to serve MANET customers in the covered area. Each user moves according to the Truncated Levy Walk mobility model [15] with a minimum speed of $1m/s$ and maximum of $1.5m/s$. We further considered a Ricean fading model, which accounts for multipath effects due to obstacles, trees and buildings.

We focus on content download to evaluate two main aspects: *(i)* the capability of CHANET to deliver the desired content to the requesting users, and *(ii)* the energetic cost, expressed in terms of Joules spent for each bit successfully delivered to the consumers. We suppose that a subset of users, randomly selected among the 25 mobile nodes in the simulated grid, are interested in downloading a set of contents provided by the AP.

We examine three cases, depending on the number of different contents downloaded from consumers: *(i) 1 CID*: all the consumers download the same

Algorithm 2. *Int-Ack* Processing

```
1: if ((HNum == MaxHops) or (SeqNum is duplicated)) then
2:    Discard the Int-Ack
3: else if (Total CoS Matching) then
4:    if (I'm the PProv) then
5:       if (Int-Ack Aggregation Check) \\Int-Ack Aggregation then
6:          Discard the Int-Ack
7:       else
8:          Send the content
9:    else
10:      if (Distance Check) then
11:         Compute the D-Object Defer Time d_d
12:         if (Counter-based Check) then
13:            Discard the Int-Ack
14:         else
15:            Send the content; \\Provider Handoff
16:      else
17:         Discard the Int-Ack
18: else if ((Partial CoS Matching) and (Distance Check)) then
19:   Compute the D-Object Defer Time d_d
20:   if (Counter-based Check) then
21:      Discard the Int-Ack
22:   else
23:      Send the D-Object(s) \\Selective Response
24:      Broadcast the modified Int-Ack
25:      Insert the modified Int-Ack in the PRT
26: else if (A matching is found in the PRT) then
27:   Discard the Int-Ack
28: else
29:   if (There is a CTP entry for the PProv) then
30:      Compute the Interest Defer Time d_i
31:      if (Counter-based Check) then
32:         Discard the Int-Ack
33:      else
34:         Broadcast the Int-Ack
35:         Insert the Int-Ack in the PRT
36:   else
37:      Discard the Int-Ack
38: return
```

content; *(ii) 2 CIDs*: two different contents are downloaded, the first one is requested by 66,66% of consumers and the second one by 33,33% of consumers; *(iii) 3 CIDs*: three different contents are downloaded, each one is requested by 33,33% of consumers. We compare the performance obtained with CHANET against that achievable by using an FTP connection over the traditional TCP/IP stack and when AODV [9] is used as the routing protocol in the MANET. To achieve fair comparison, TCP Vegas has been implemented because of its higher

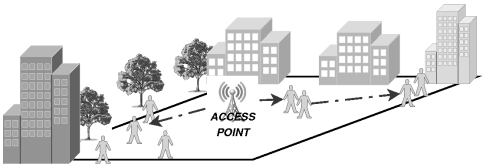


Fig. 3. Simulation scenario

performance in MANETs compared with other TCP versions [16]. It detects incipient congestion by monitoring variations of the packet delay (instead of losses).

While in the FTP over TCP scenario contents can be downloaded only through the AP, in CHANET content chunks can be also obtained by any node storing them. In both cases, we consider IEEE 802.11g as the access technology. Main parameters are reported in Table 1 as taken from the standard and from datasheets of commercially available devices [17]. Transmission power and receive sensitivity values of the simulated devices are used as inputs to the ns-2 energy model. At the beginning of the simulation, each node has assigned an initial energy level that is decremented at any packet transmission, reception and overhearing. We set the initial energy very high to be sure that no node runs out energy during the simulation.

Parameters concerning CHANET and the legacy protocol stacks are summarized in Table 2.

Table 1. 802.11g Simulation Parameters

PHY Parameter	**Value**
Frequency	2.4 GHz
Receive Sensitivity	-86 dBm
Transmission Power	18 dBm
Power consumption while transmitting	1.74 W
Power consumption while receiving	0.9306 W
Power consumption while idle	0.6699 W
MAC Parameter	**Value**
SlotTime	9 μs
SIFS	10 μs
Preamble Length	96 bit
CWmin - CWmax	15 - 1023
Short - Long Retry Limit	3 - 7
MAC header	34 bytes

We evaluate both network performance metrics and energy cost metrics. Concerning the former, we consider the *Download Time* as a good indicator of the user experience. It is defined as the average time required for a user to download the requested content, which in our case is assumed to be a 10MB file. Also signalling overhead is taken into account in the network performance. Specifically, we compute two types of overhead, respectively related to the consumer and to the overall network. *Consumer Overhead* is defined as the average ratio between

Table 2. Architectures' Simulation Parameters

CHANET	Value	TCP/IP/AODV	Value
Interest size	24 bytes	TCP Vegas header	20 bytes
Int-Ack size	40 bytes	TCP Vegas α	1 packet
D-Object header size	40 bytes	TCP Vegas β	3 packets
D-Object payload size	1000 bytes	TCP Vegas γ	1 packet
Chunk size	10 Kbytes	Payload size	1000 bytes
Aggregation time	20ms	AODV RREQ size	48 bytes
MaxHops	10	AODV RREP size	44 bytes
Defer time (Data)	[*SlotTime, CWmin*SlotTime*]	AODV RERR size	32 bytes
Defer time (Interest, Int-Ack)	[*CWmin*SlotTime, 2*CWmin*SlotTime*]	AODV HELLO	disabled

the transmitted signalling bytes and the received data bytes for a consumer. It represents a measure of efficiency from the consumer point of view. In the computation, only the signalling originated by consumers is considered, packet duplications in the network are not included. CHANET signalling packets include *Interest* and *Int-Ack*; no further overhead is taken into account, since at the MAC layer no retransmission is allowed due to the broadcast nature of all exchanged packets. For the TCP/IP case, the signalling overhead includes TCP control packets (three-way handshaking segments and all ACKs transmitted by consumers to the AP), AODV control packets (route request, route reply and route error packets), and the MAC acknowledgments for unicast data transmission (Request-To-Send/Clear-To-Send exchange is disabled). The *Network Multiplication Factor* aims to quantify the percentage of bytes totally generated into the network per each received data byte. It is defined as the ratio between all the bytes (signalling and data) sent by all nodes (source, consumers and forwarders) over the MANET and the data bytes received by all consumers. It gives a measure of the "multiplication" factor of the network since it takes into account packet duplications and retransmissions. Concerning the energy cost, we compute it in terms of *Energy per Bit*, which is defined as the Joules spent for each bit successfully delivered to the consumers. Energy consumption is evaluated for all the involved parties: network, consumers, and network operator. The *Network Energy Cost* is an overall cost parameter defined as the energy consumed by the whole network (source, consumers and all involved nodes in the MANET) to deliver the total amount of required bits to the consumers. The *Consumer Energy Cost* is the mean energy spent by a consumer to receive a bit. The *Operator Energy Cost* is defined as the energy spent by the source for each bit delivered to a consumer. When the source is an AP managed by a network operator, this energetic cost has a direct impact on the operational expenditures (OPEX).

Figure 4 shows the performance in terms of Download Time. We observe that CHANET outperforms the legacy approach thus achieving faster downloads. The delay increases with the number of consumers due to higher load and congestion, but in CHANET this trend is smoother than for traditional FTP. Motivations

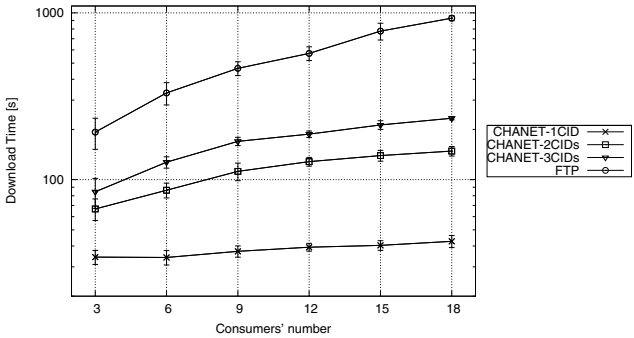


Fig. 4. Download Time

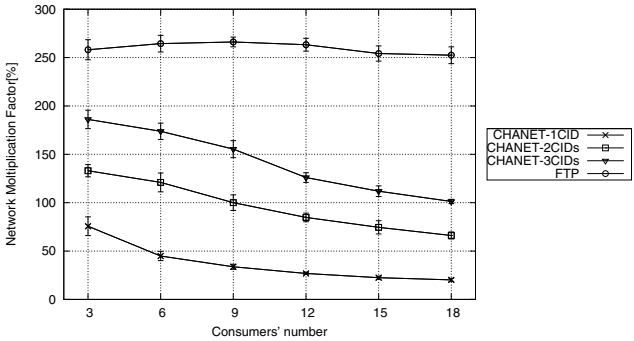


Fig. 5. Network Multiplication Factor

of such an advantage may be found in both Figures 5 and 6. In Fig. 5 the Network Multiplication Factor is plotted against the number of consumers. With CHANET, the network load does not increase with the consumer number, like in the legacy suite case, rather it decreases, since a single transmission may deliver data to more users simultaneously. This effect is amplified with the number of receivers. The *Consumer Overhead* is also very low compared to FTP, as shown in Fig. 6. Of course, CHANET performance get worse with the increasing of the number of CIDs, since the probability that a single packet transmission serves more than one user decreases.

Figures 7 - 9 summarize the protocol behaviours with respect to the energy efficiency. CHANET is significantly better performing than the legacy case. Under some circumstances (i.e., a single CID case) the difference between the two cases is of an order of magnitude. Figure 7 reports the Network Energy Cost that is the most practical indicator of the pollution produced by transmissions. Not only CHANET is more efficient than TCP/IP, but its efficiency also increases with the number of consumers. The reason is again due to the capability of CHANET to serve more users in a single transmission, thus reducing the total number of transmissions in the network (see Fig. 5) and avoiding unnecessary

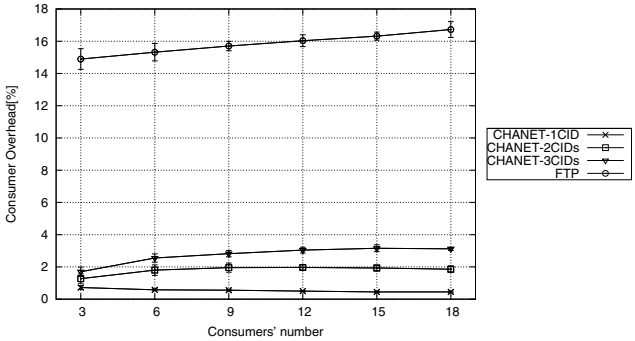


Fig. 6. Consumer Overhead

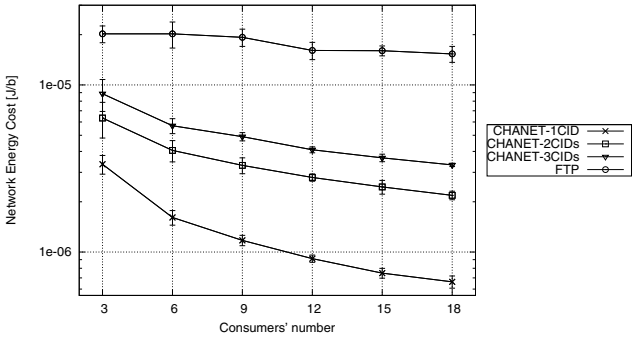


Fig. 7. Network Energy Cost

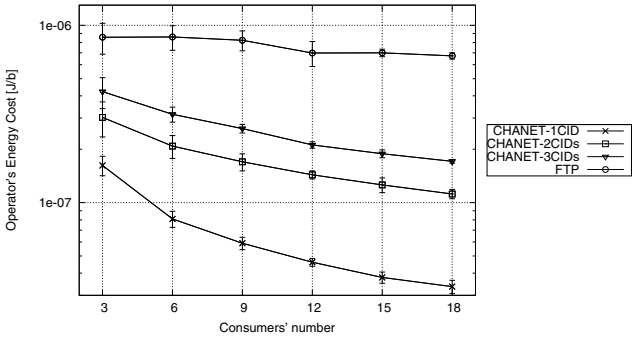


Fig. 8. Operator Energy Cost

energy consumption. Once again the efficiency worsens with the increasing number of CIDs as a direct consequence of the network load. The Operator Energy Cost represented in Fig. 8 shows that CHANET outperforms the legacy suite by ensuring to the network operator a much higher efficiency. As in the previous case, the efficiency increases with the number of consumers. From an economical point,

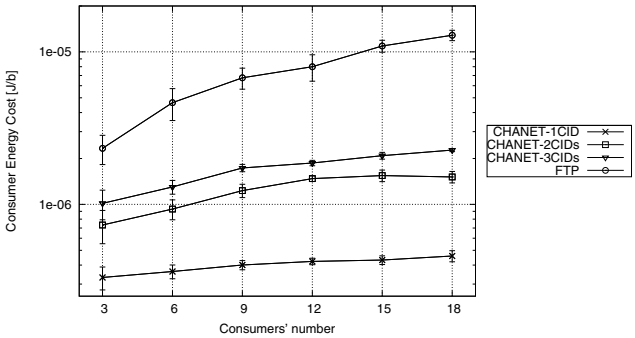


Fig. 9. Consumer Energy Cost

we may say that an operator who decides to run CHANET on its APs may get a substantial reduction in the OPEX cost. Finally, Fig. 9 reports the *Consumer Energy Cost*. Also in this case, CHANET performs better than TCP, but we observe a different trend when varying the number of CIDs. When a single CID is downloaded by all the consumers, the energy cost is lower. This is due to the packet overhearing, which allows a consumer to receive data without even requesting it and, hence, without spending energy to request it. However, overhearing becomes less effective at the increasing of the number of CIDs, since the effects of packets collisions and retransmissions prevail. By considering, for instance, the case of 2 CIDs, overhearing is not effective from 3 to 9 consumers and effects of collisions prevail so energy costs increase, while, for 12 to 18 consumers, overhearing starts to be effective again and overcomes the adverse effects of collisions.

5 Conclusions

In this paper, we developed a new content-centric energy-efficient architecture named CHANET that achieves content retrieval, delivery and caching in IEEE 802.11 MANETs. Simulation results show the great benefits offered by CHANET in terms of higher energy efficiency, reduced latency and control overhead compared to traditional MANETs based on the TCP/IP suite.

References

1. IEEE Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, Std. 802.11-2007 (June 2007)
2. Jacobson, V., et al.: Networking Named Content. In: ACM CoNEXT, Rome, Italy (December 2009)
3. Pan, J., Paul, S., Jain, R.: A Survey of the Research on Future Internet Architectures. Communications Magazine 49(7), 26–36 (2011)
4. Ahlgren, B., et al.: A Survey of Information-Centric Networking. Dagstuhl Seminar (February 2011)

5. Meisel, M., Pappas, V., Zhang, L.: Ad Hoc Networking Via Named Data. In: ACM MobiArch 2010, Chicago, Illinois (September 2010)
6. Meisel, M., Pappas, V., Zhang, L.: Listen First, Broadcast Later: Topology-Agnostic Forwarding Under High Dynamics. In: Annual Conference of International Technology Alliance in Network and Information Science, London, UK (September 2010)
7. Oh, S.Y., Lau, D., Gerla, M.: Content Centric Networking in Tactical and Emergency Manets. In: IFIP Wireless Days, Venice, Italy, pp. 1–5 (October 2010)
8. Tonguz, O., et al.: On the Broadcast Storm Problem in Ad Hoc Wireless Networks. In: Broadband Communications, Networks, and Systems (BROADNETS), San Jose, CA (October 2006)
9. Perkins, C.E., Belding-Royer, E.M., Das, S.: Ad Hoc on Demand Distance Vector (AODV) routing. IETF, RFC 3561 (July 2003)
10. Jones, C.E., et al.: A Survey of Energy Efficient Network Protocols for Wireless Networks. Wireless Networks 7(4), 343–358 (2001)
11. Lee, U., Rimac, I., Hilt, V.: Greening the Internet with Content-Centric Networking. In: e-Energy 2010. University of Passau, Germany (2010)
12. Lee, U., Rimac, I., Kilper, D., Hilt, V.: Toward Energy-Efficient Content Dissemination. IEEE Network 25(2), 14–19 (2011)
13. Intanagonwiwat, C., et al.: Directed Diffusion for Wireless Sensor Networking. IEEE/ACM Transactions on Networking (TON) 11(1) (February 2003)
14. The Network Simulator-2 (ns-2), http://www.isi.edu/nsnam/ns
15. Rhee, I., et al.: On the Levy-Walk Nature of Human Mobility. In: IEEE INFOCOM, Phoenix, AZ (April 2008)
16. Papanastasiou, S., Ould-Khaoua, M.: Exploring the Performance of TCP Vegas in Mobile Ad Hoc Networks. International Journal of Communications Systems 17(2), 163–177 (2004)
17. Cisco aironet 802.11a/b/g wireless cardbus adapter. Data Sheet available on line at http://www.cisco.com/en/US/prod/collateral/wireless/ps6442/ps4555/ps5818/product_data_sheet09186a00801ebc29.pdf