Cloud-Integrated WOBAN: An offloading-enabled architecture for service-oriented access networks

Abu (Sayeem) Reaz [a], Vishwanath Ramamurthi [a], Massimo Tornatore [b,a,*], Biswanath Mukherjee [a]

[a] *University of California, Davis, USA*
[b] *Politecnico di Milano, Milan, Italy*

Received 14 May 2013
Received in revised form 9 November 2013
Accepted 4 December 2013
Available online 28 February 2014

1. Introduction

Recent industry research trends [1] show that wired and wireless networking technology must be treated as an integrated entity to create a flexible, service-centric network architecture. A possible such architecture is a wireless-optical broadband access network (WOBAN) (Fig. 1) that is formed by a wireless mesh network (WMN) to provide end users access and an optical backhaul network to carry the aggregated traffic collected over the WMN [2]. Improvement of wireless technologies, ease of deployment, and cost-effectiveness has made WMN an attractive access network architecture for real-world deployment [3]. The optical backhaul network comprises of a Passive Optical Network (PON) [4] with an Optical Line Terminal (OLT) in the Central Office (CO) which is connected via optical fiber

* Corresponding author at: Politecnico di Milano, Milan, Italy. Tel.: +39 02 2399 3683; fax: +39 02 2399 3413.

E-mail addresses: asreaz@ucdavis.edu (A. Reaz), rama@ucdavis.edu (V. Ramamurthi), tornator@elet.polimi.it (M. Tornatore), bmukherjee@ucdavis.edu (B. Mukherjee).

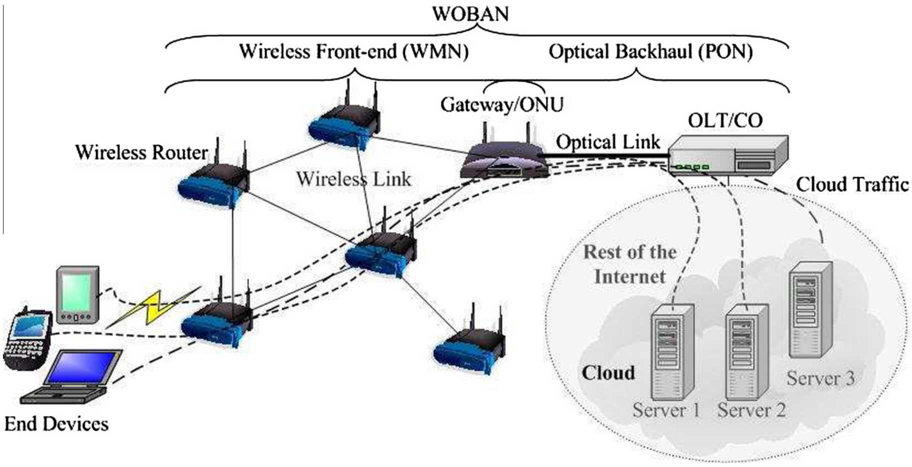

Fig. 1. WOBAN architecture and its traditional cloud access.

to multiple Optical Network Units (ONU). At the front-end of WOBAN, a set of wireless nodes (routers) form a WMN. A selected set of these nodes, called gateways, are connected to the optical part of the network. In this way, a WOBAN can achieve cost-effective deployment of a WMN while having higher performance due to the optical backhaul network. Furthermore, a centralized OLT allows traffic in the wireless and optical segments of a WOBAN to be managed together [5]. Such benefits make such an architecture commercially attractive [7].

Today's access network proposals, such as WOBAN, are being increasingly shaped by the services that they provide to the end users. Currently, in a WOBAN, cloud services are accessed from cloud servers that can be anywhere in the metro/core network. Consequently, a user's request for a cloud service must be transported through the wireless mesh, possibly over multiple hops, to reach the OLT via the gateways and be delivered to an appropriate server. The response from the server for the different services are also transported over possibly multiple wireless hops to the end users. In this *traditional setting* (Fig. 1), the services requested by a user may consume a significant amount of wireless bandwidth in the WMN, which is limited because of contention, interference, and limited capacity [8]. Moreover, because most traffic flows have to go through one of the few gateways, links near the gateways become a bottleneck. In this paper, we propose a service-oriented architecture, called Cloud-Integrated WOBAN (CIW), to address these limitations by creating an integrated platform to provide cloud services.

In CIW, we propose to *integrate* cloud components (CC), such as storage [9] and servers [10], within the WMN of a WOBAN. A CC can be a storage and/or processing unit that can provide services. Many cloud services are local and the geographic relevance of these services is within the footprint of the front-end of a WOBAN. For example, finding parking locations, parking rates, and parked cars can be done with local information. Unlike a typical content distribution system, a CIW hosts and serves these cloud services locally using the CCs. These services can be accessed from the CCs through web-based interfaces [11].

CCs not only can store contents, but can also facilitate the services associated with these contents. Note that a CC can periodically synchronize with a centralized cloud (e.g., a data center) for distributed service availability. Availability of these low-cost and easy-to-deploy CCs [9,10] make CIW a technologically feasible and practically deployable architecture. Even though resources of these CCs are limited compared to a centralized cloud, each can have sufficient processing and storage capability [9,10] to host and serve a certain (limited) number of local services. Such an architecture allows the traffic associated with these cloud services to be offloaded from the wireless backhaul and diverted towards the CCs. This reduces the bandwidth usage of the wireless front-end of WOBAN, removes traffic bottleneck near gateways, increases carried load of WOBAN, and facilitates local updates to the cloud, as shown in Fig. 2. Hence, deployment of CCs saves some of the bandwidth needed for the local services at the "cost" of processing and memory capabilities introduced in the WMN of a WOBAN.

Integrating a cloud with WOBAN imposes a few unique challenges that are unlike traditional cloud over network approaches. As shown in Fig. 2, CIW should be specifically designed so that it can *offload* the traffic from the wireless backhaul of WOBAN to the locally-hosted CCs. As offloading is currently being considered as a technique to increase scalability and reduce bottleneck in the backhaul [12], this inherent offloading property of CIW makes it an attractive architecture. Offloading of service traffic decreases wireless contention, reduces congestion near the gateways, and frees up capacity for other traffic which can be used to increase the carried load of the network. The other challenge is the traffic pattern in a WOBAN where almost all the service traffic

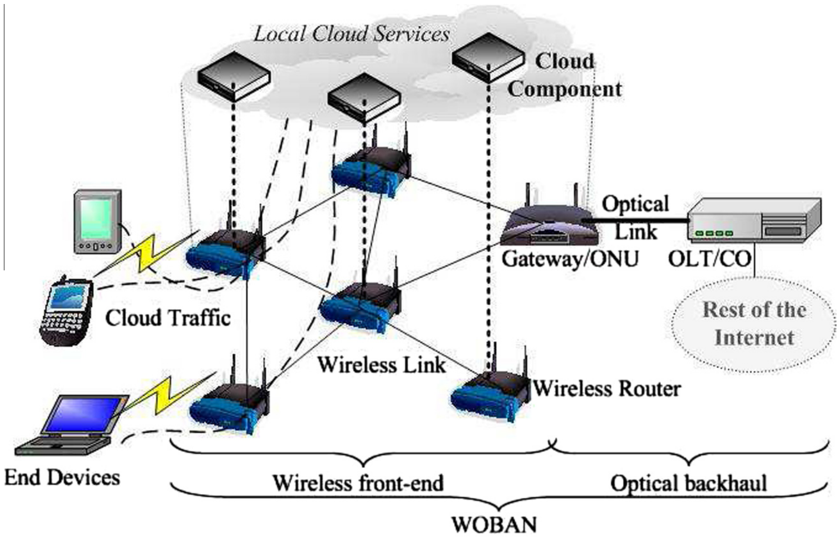

Fig. 2. A Cloud-over-WOBAN (CIW).

travels to and from the OLT. CIW should be designed to redistribute this traffic across the network which achieves further performance improvement for WOBAN.

To manage the traffic efficiently, CIW needs an algorithm that carries the local cloud traffic to the CCs while routing the other traffic through the OLT. Moreover, as the CCs host local services, the demand for such services may vary depending on the locality and the time of the day, and idle CCs can be turned off for energy savings. In the same way, idle ONUs can be turned off as well when load is low [13]. To turn off a network components, the typical energy-savings approach is to pack the flows on as few links as possible [13]. Because of the packing of flows, performance of WOBAN may degrade for this scheme. Hence, we propose a new scheme, Green Routing for CIW (GRC), that *self-manages* the network components by activating a minimum number of CCs and ONUs while shutting down the unused ones in CIW to achieve power savings. It also performs "anycast" load-balanced routing through the active ONUs and CCs using shortest-delay paths utilizing the architecture of a PON. Note that GRC can be directly implemented on WOBAN even without the integrated cloud and power-saving modes.

In this paper, we address both design and operational aspects of a CIW. First, for the design aspect, we discuss the architecture and implementation options for a CIW. We also present a placement scheme for CCs and their services in a CIW through a Mixed-Integer Linear Program (MILP). For the operational aspect, we present a self-managing green routing heuristic algorithm, GRC, and discuss how it achieves both the energy savings and routing for a CIW. Our performance evaluation shows that CIW architecture increases the carried load of WOBAN. We also show that GRC achieves a significant power savings while maintaining a lower average packet delay in a CIW. Our performance evaluation also supports designing GRC as a heuristic solution because GRC can favorably operate under different network scenarios.

The rest of the paper is organized as follows. Section 2 discusses the related work. Section 3 introduces CIW. Section 4 presents optimal placement of cloud components in a CIW. Section 5 discusses the GRC algorithm and how to achieve energy savings through it. Illustrative numerical examples are shown in Section 6. Section 7 concludes the study.

2. Related work

In this work, we propose a service-oriented architecture (SoA) for WOBAN. We also introduce a green routing algorithm that achieves energy savings for WOBAN. Hence, we briefly discuss some related works in the literature for each of the aspects of our work, namely SoA and energy-savings in wireless and optical access networks.

- WOBAN*:* there are several proposals for performance improvements for WOBAN. Capacity and Delay-Aware Routing Algorithm (CaDAR) [5] performs optimum capacity distribution on wireless links and performs shortest-delay-path routing in WOBAN. Cost-effective capacity enhancement of the WMN is shown in [8] to improve the performance over wireless backhaul. However, none of the solutions integrates cloud with WOBAN for a service-oriented design.
- *SoA:* there are several papers in the literature that discuss network architectures designed for services. An overview of how web services can be provided using SoA is presented in [11]. In [14], a traditional view of SoA is presented where service is provided through web-based interfaces. In another similar approach,

Ref. [15] promotes how information technology can generate value in a SoA. In [16], a service-oriented network design is proposed where different components of SoA exchange information through a network. A service-oriented network design for wireless sensor networks is presented in [17]. Many of the services requested by the users are local; hence, location-dependent services are important to identify. An overview and classification of services that are location dependent are discussed in [18]. It opines that, for any wireless access network such as a WOBAN, location-based services, i.e., local services, are an additional source of revenue generated from the investments in the infrastructure [18]. However, none of them discusses an infrastructure-based integration of cloud services with an access network.

- *Facility location problems:* in general, a facility location problems in a telecom network involve placement of facilities at most suitable locations. Suitability of a location is usually determined based on the objective (e.g., network bandwidth, access latency, etc.) of the problem. Placement of service components in network sites is discussed in [19], which presents four algorithms for placement of services based on the cost of acquiring components to be installed at the various sites. In [20], a solution approach for determining number and locations of service facilities in a distributed and scalable manner is proposed. This approach includes iterative re-optimization of the number and locations of the service facilities within predetermined locations. A generic discussion based on current and suggested practices for content distribution (i.e., facility location for contents) in presented in [21], which elaborates content distribution practices associated with different data mining approaches and their corresponding benefits. Different challenges in providing effective ways of using these practices for large-scale data management involved in content distribution is further discussed in [21]. Even though facility location problems are discussed in these works, they do not consider specific wireless characteristics of a WMN. As a result, they do not address the specific set of challenges that are unique to wireless backhaul of a WOBAN.
- *Energy-savings:* various energy-saving approaches have been proposed for cloud-enabled access networks. A relevant proposal to our work is the utilization of nanodata centers (NaDa) to lower the energy consumption in the network, as proposed in [22], that integrates small-scale content delivery system at the edge of the network and uses peer-to-peer (P2P) communication for content distribution. However, such implementation introduces an additional burden on the end users with additional power and bandwidth requirement. An energy-saving routing scheme for WOBAN is proposed in [13] that packs flow on fewer links to reduce the load on ONUs so that they can be shut down for energy saving. Because of the packing of flow, the performance of WOBAN may degrade for this scheme. Moreover, this scheme does not consider cloud services.

  Focusing specifically on the *optical* domain, there are several proposals on energy efficiency for PON. A low-power state for PON equipment has been proposed in [23]. Challenges and solutions to put ONUs into low-power mode for power saving using MAC-layer control and scheduling is discussed in [24].

  In the *wireless* domain, there are numerous proposals for energy savings, e.g., a model to represent the performance of multi-hop radio networks under energy constraints is presented in [25]. It also presents routing algorithms that can intelligently utilize the available energy. In [26], the authors propose energy-aware routing policies that find minimum-energy routes while avoiding energy depleted nodes. A two-layer wireless sensor network architecture is proposed in [27] to maximize the lifetime of sensor nodes. However, as wireless nodes in a WMN provide user access, they usually collect, inject, and forward user traffic. As a result, most of the wireless nodes, especially ones near the gateways, need to be active most of the time and finding energy-efficient paths over the wireless mesh is often difficult.

3. Cloud-Integrated WOBAN (CIW)

In order for a service provider to guarantee the cloud services to its users, it is important to integrate the services into network provisioning and planning. Hence, we present CIW, an integrated infrastructure platform to provide cloud services within an access network.

3.1. Motivation for CIW

Many of the cloud services requested by the end users are local, and quite often, it is possible to serve them locally. For example, finding parking locations, rates, and finding parked cars can be served with local information; spam originating from a WOBAN can be blocked within the wireless front-end without the need for it to go to the OLT. For such local services (e.g., parking location), it is better to manage the local changes locally. Hence, we consider how to integrate a cloud with WOBAN to create a CIW. It offloads traffic from wireless backhaul and reduces its bandwidth usage, removes bottleneck near gateways by diverting traffic away, increases carried load of the network by responding to requests locally, and keeps updates local, as shown in Fig. 2.

It is *important* to note that integrating a cloud is particularly suitable for a WOBAN. As WOBAN is driven by an OLT from a central office, it is possible to route and manage the wireless and optical traffic together. With today's technology, memory and processing power is available within reasonable cost [10]. On the other hand, wireless backhaul bandwidth is scarce [28]. CIW can address these limitations of WOBAN in a cost-effective way.

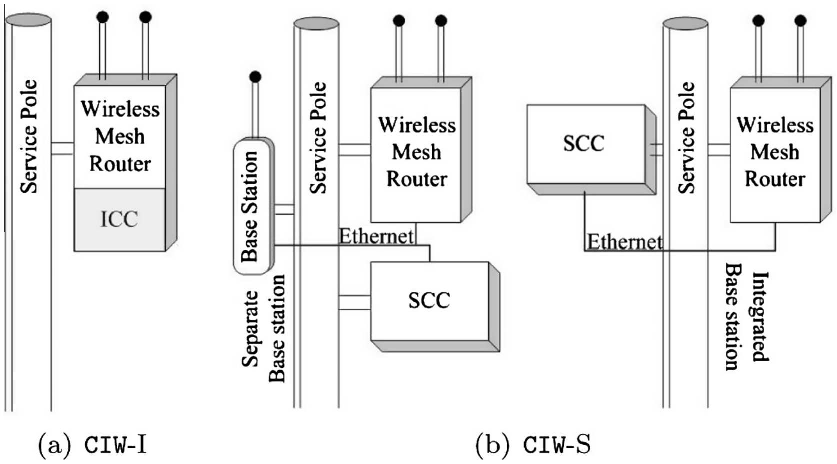

Fig. 3. Various implementation approaches for a CIW.

3.2. CIW *architecture*

In a CIW, we add CCs, such as storage, or a server, with the wireless nodes of a WOBAN. If a cloud-service request can be served from within, it is forwarded to the appropriate CC instead of the OLT. Then, the CC sends back the corresponding response. Depending on the policy of the service provider, equipment provider, and types of services, these CCs may differ. If a CC is deployed with a wireless node, we refer to it as a host node for services.

When an end user needs to access a cloud service from a CIW, a cloud request is sent to the wireless node it is connected to. When the request reaches the wireless node from the user, if the service is available from a CC attached to that wireless node, it directly responds to the end user after processing. If the service is not available, the wireless node sends the request to a nearby host node if the service is available from the WOBAN. Otherwise the request is sent to the OLT.

These are some of the possible options of implementing a CIW.

1. CIW-I: If there is user development space on WMN products, and a blank wireless box, where a developer has the flexibility to implement and deploy all the operational aspects of a wireless router, is available, then each CC can be integrated within the wireless node. This implementation model can be developed over Wisper [29] WMN products. For this Integrated CC (ICC) based implementation, additional hardware need may be minimal as each node has some spare processing capacity and memory that can be utilized to host services. In that case, the cost of bandwidth is more important than hardware cost. But, in such an implementation, each node may host only a few cloud services, so the services should be distributed among all the nodes. This design is called CIW-I as shown in Fig. 3(a).

2. CIW-S: For other implementation models, additional Selective CCs (SCC), e.g., a “light weight” or “thin” server [10], or a storage [9], with capability to serve local cloud services, are connected via Ethernet to some wireless nodes. This model is suitable for WMN products where additional development space is not available and may be more common. For example, a wireless node from Firetide [30] consist of a base station (BS), which is used to communicate with the end users, and a wireless router, which is used to form the wireless backhaul of the WMN, as shown in Fig. 3(b). On the other hand, in wireless nodes from Tropos [31] and Meraki [32], user access and wireless backhaul formation are integrated (Fig. 3(b)) and each wireless node is equipped with at least two radios: one for end user access and one for forming the wireless backhaul. For this implementation, as additional SCCs are deployed, each of them can potentially host several services, so only a few such server is needed. The design goal is to deploy as few SCC as possible while serving all the desired services. This design is called CIW-S as shown in Fig. 3(b).

4. Optimum placement of CCs and services in CIW

In a CIW, a set of wireless nodes should be equipped with CCs. These CCs are distributed based on the demands and requirements of the cloud services. Which services are to be hosted by which CC is also to be determined. To have a cost-effective solution, the number of CCs should be minimized in such a way that it increases the bandwidth efficiency while hosting all the services. Here, the “cost” of each service has two parts: (a) cost of processing and memory and (b) cost of bandwidth. Processing and memory costs of a service depend on the processing and memory requirement for a CC to host that service. In our design, we incorporate processing and memory requirements of each service and their corresponding costs to ensure that each CC has sufficient memory and processing capability to serve the services it may host. This allows a network designer to consider these design issues and provides the flexibility to determine the costs and requirements of each service depending on specific design criteria. Note that, for cost of bandwidth, we only consider the wireless backhaul capacity, where we intend to gain our performance improvement. We do not consider user access because usually each node has dedicated resources (e.g., a dedicated radio) to communicate with end users.

Our *objective* is to minimize the deployment cost of CCs in CIW, under the *constraints* that hosts for each service should be reached within a number of certain hops, and the number of services at each CC is within it's memory and processing capability. We develop the following MILP to formulate our problem.

4.1. Inputs to the MILP

- $\omega(N)$ = set of N nodes of the WMN of a WOBAN; locations of these N nodes are given,
- $\psi(K)$ = set of K services to be deployed over WOBAN,
- h_C = maximum number of hops a service request/response travels over the *wireless backhaul.*
- C_u = radio capacity of node u
- I_{uv} = set of links that interfere with link (u, v)[1]
- α_{uv} = fraction of background traffic (regular mesh traffic) on each link (u, v)
- D_u^I = cost of an SCC deployment at node u
- D_m^k = cost of memory for service k
- D_p^k = cost of processing for service k
- D_{BW}^k = cost of unit flow over a link for service k
- M = a large number
- R_k^M = memory resource required for service k
- R_k^P = processing resource required for service k
- ϕ_u^M = memory resource available at node u
- ϕ_u^P = processing resource available at node u
- γ_i^k = aggregated demand for service k from node i
- $z_{u,h}^{k,i}$ = 1, if demands for service k from node i are served by node u within h_C hops

4.2. Notations for the MILP

- u = a node in WMN of a WOBAN, $u \in \omega(N)$
- v = a neighbor of node u in WMN of a WOBAN, i.e., node u has a transmission link to node v, $v \in \omega(N)$
- k = a cloud service to be deployed in WOBAN, $k \in \psi(K)$
- i = a node in WMN of a WOBAN with a demand for a cloud service, $i \in \omega(N)$

4.3. Variables for the MILP

- x_u^k = 1, if service k is served from node u
- y_u = 1, if an SCC is deployed at node u
- $w_u^{k,i}$ = 1, if service k is served from node u for a demand from node i
- $\lambda_{vu}^{k,i}$ = flow on link (u, v) for service k for a demand from node i
- C_{uv} = capacity on link (u, v)

$$Minimize: \sum_k D^k + \sum_u y_u * D_u^I, \tag{1}$$

$$D^k = \sum_u x_u^k * \left(D_m^k + D_p^k\right) + \sum_i \sum_u \sum_v \left(\lambda_{uv}^{k,i} * D_{BW}^k\right); \qquad \forall k \tag{2}$$

$$y_u \geqslant \frac{\sum_k x_u^k}{M}; \qquad \forall u \tag{3}$$

$$\sum\nolimits_k \left(x_u^k * R_k^M\right) \leqslant \phi_u^M; \qquad \forall u \tag{4}$$

$$\sum\nolimits_k \left(x_u^k * R_k^P\right) \leqslant \phi_u^P; \qquad \forall u \tag{5}$$

$$\sum\nolimits_u \left(x_u^k * z_{u,h}^{k,i}\right) \geqslant 1; \qquad \forall k, \quad \forall i \tag{6}$$

$$\sum\nolimits_v \lambda_{vu}^{k,i} - \sum_v \lambda_{uv}^{k,i} = \gamma_i^k * w_u^{k,i}; \qquad \forall k, \quad \forall u, \quad \forall i \ (i \neq u) \tag{7}$$

$$w_u^{k,i} \leqslant x_u^k; \qquad \forall u, \quad \forall i, \quad \forall k \tag{8}$$

$$\sum\nolimits_u w_u^{k,i} = 1; \qquad \forall k, \quad \forall i \tag{9}$$

$$\sum\nolimits_i \sum_k \lambda_{uv}^{k,i} \leqslant (1 - \alpha_{uv}) * C_{uv}; \qquad \forall u, \quad \forall v \tag{10}$$

$$\sum\nolimits_v C_{uv} + \sum_v C_{vu} \leqslant C_u; \qquad \forall u \tag{11}$$

$$\sum\nolimits_v C_{vu} + \sum_{(p,q) \in I_{uv}} C_{pq} \leqslant C_u; \qquad \forall u \tag{12}$$

$$x_u^k, \quad y_u, \quad w_u^{k,i} \in \{0, 1\}; \qquad \forall k, \quad \forall u, \quad \forall i \tag{13}$$

The objective function, Eq. (1), minimizes cost for all services. The first part of Eq. (1) consists in the sum of processing, memory, and bandwidth cost for each service. The second part of Eq. (1), expressed by D_u^I, represents the cost for CC's deployment and it is applicable only if CIW-S is deployed. For CIW-I, all the values of D_u^I are 0. Among the constraints, Eq. (2) defines the cost of each service: the first part indicates the processing and memory cost of each cloud service, and the second part describes the cost of bandwidth. If a wireless link (u, v) carries some traffic for any service k, then it will incur a bandwidth cost, otherwise it will not. The bandwidth cost in Eq. (2) is linear to the flows on the link because it ensures that the total bandwidth cost incurred on a link is proportional to the flow on the link. Eq. (3) indicates if an SCC is deployed at a node u. Eqs. (4) and (5) bound the processing capacity and memory at each node such that the total processing and memory need for all services at any node u should be less than its available processing capacity and memory. Note that, as y_u is binary, it chooses the locations of the SCC. If more than one SCC per location are to be deployed, it can easily be done by allowing more processing capacity and memory using Eqs. (4) and (5). Hence, it provides a designer with flexibility of applying the MILP for different scenarios.

Eq. (6) bounds the number of hops for each cloud service such that each node that requests any service k should be covered by at least one host node within h_C hops over the wireless backhaul. The value of h_C should not be large (e.g., more than 3) because the performance of WMN de-

[1] Let SNR_{uv} = signal-to-noise ratio of link (u, v), P_{uv} = transmitted power over link $(u, v), G_{uv} = distance(u, v)^a, a$ = polynomial constant, σ = thermal noise at receiver, β = threshold for SNR_{uv}, S = set of links s.t. $k \neq u, \ v, \ x \neq v$. A link $(p, q) \in I_{uv}$ if $SNR_{uv} = \frac{P_{uv} G_{uv}}{\sum_{(k,v) \in S} P_{kx} G_{kv} + \sigma^2} \leqslant \beta$ when link (p, q) is active [28].

grades with a larger number of wireless hops. This constraint holds when, for any service k, at least one of the nodes that is reachable from node i within h hops hosts service k and the corresponding indicator variable x_u^k takes the value 1. Eq. (7) enforces the *flow constraint*. If a request from any node i for service k is terminated at node u, then the demand γ_i^k is served at u. Compared to the classical flow constraint in network design problems, here the destination is not known in advance; it can be any of the host nodes of service k, including node i. The variable $w_u^{k,i}$ determines if node u is the destination for a demand from node i. If node u is the destination, the flow constraint takes up the value γ_i^k, otherwise 0. Moreover, Eq. (7) ensures that, if node i hosts service k, then there is no flow introduced to the network for a demand from node i. Note that, in a dynamic scenario, γ_i^k is likely to vary. However, for the network design problem, we consider the peak demand that can be obtained from empirical data.

Eq. (8) ensures that a request from any node i for service k is served by one of the host nodes of service k. Eq. (9) ensures that only one node u serves a request from node i for service k. Eq. (10) is the *capacity constraint*: it ensures that the service traffic is within the capacity that is not used by background traffic. Eqs. (11) and (12) are specific to wireless capacity assignment [28]. Eq. (11) describes that a wireless node's transmission capacity is shared with the reception capacity, as a wireless radio cannot transmit while receiving. Similarly, Eq. (12) describes that the reception capacity of a wireless node is shared with all the links for which the signal-to-noise-and-interference ratio (SINR) is higher than a threshold. Eq. (13) defines binary indicator variables for each node. If any node u should host service k, the value of x_u^k is 1, otherwise the value is 0. Similarly, the value of $w_u^{k,i}$ is 1 if a request from any node i for service k is served by node u.

5. Green routing for CIW (GRC)

After the design of a CIW, we present its the operational aspects. For efficient operations and provisioning, we present a heuristic algorithm that self-manages network components and traffic to achieve power savings and load-balanced routing. Note that we design GRC as a heuristic solution because, through our performance studies in Section 6.2, we observe that GRC can successfully address the specific issues that it is designed to address under different network scenarios.

5.1. Routing in GRC

In this section, we address various aspects of routing over CIW such as efficient path computation for both cloud traffic (i.e., traffic to and from integrated and centralized cloud) and regular traffic (i.e., non-cloud traffic, such as point-to-point traffic).

5.1.1. Auxiliary graph formation

For our routing algorithm, we first represent CIW as an auxiliary graph, as shown in Fig. 4: all the horizontal links are the routing links (wireless and optical links) and the vertical links are "virtual" links that assist the anycast routing. Among the vertical links, L_C links indicate if a CC is hosted at a wireless node and L_S links indicate if a cloud service is hosted at a particular CC. For example, an L_C link (u, CC_1) means node u hosts CC_1. Similarly, an L_S link (S_1, CC_1) means service S_1 is available at CC_1. L_D links indicate if a service is hosted in CIW and allow GRC to perform anycast routing among the active CCs. Among the horizontal links, L_Ws are wireless links for every wireless node pair within transmission range of one another. L_Es are Ethernet links between a GW and an ONU and L_Os are optical links between ONUs and the driving OLT. Using appropriate weight assignment over these links, GRC can achieve shortest delay and green routing.

5.1.2. Link-state advertisement

In GRC, each wireless node advertises the link-state advertisement (LSA) for each link in L_W and the OLT broadcasts LSA for each link in L_O periodically. Gateways advertise the LSA for both L_W and L_E it is incident to. Depending on the load and capacity on each link, GRC calculates the delay on each routing link, assuming independent arrivals and assigns weights to the links based on their delays. Links with higher delay are assigned higher weights and vice versa. Note that, it is shown in [5] that LSAs have low dispersion time in a WOBAN.

5.1.3. Path computation for regular traffic

For regular traffic, all the traffic goes through the OLT. So, for upstream traffic, GRC computes a path to the OLT at each wireless node, allowing the traffic to travel through *any* of the active ONUs using the shortest-delay path. As a result, GRC allows a wireless node to select the most suitable path through anycast routing and balance load across active ONUs. Similarly, for downstream traffic, it computes a path to the destination wireless node at the OLT. Thus, the shortest-delay path is computed for each source–destination $(s-d)$ pair over $\{L_W \bigcup L_E \bigcup L_O\}$ at node s.

If C_{uv} is the capacity and λ_{uv} is the flow on link $(u,v) \in L_W$, then the weight on link (u,v), W_{uv}^W, is calculated as $\frac{1}{C_{uv}-\lambda_{uv}} + \frac{1}{2C_{uv}}$, where $\frac{1}{C_{uv}-\lambda_{uv}}$ is the queuing and precessing delay, and $\frac{1}{2C_{uv}}$ is the synchronization delay for TDM-based operation of a wireless channel [5].

Similarly, weight W_{uv}^E for link $(u,v) \in L_E$ is calculated as $\frac{1}{C_{uv}-\lambda_{uv}}$. If the ONU is inactive, then $W_{uv}^E = \infty$. If P_{uv} is the propagation delay between the OLT and ONU O, $D_{\tilde{O}}$ is the upstream delay on all ONUs other than ONU O in the PON system (can be calculated by the OLT from dynamic bandwidth allocation), and D_W is the average waiting time for data destined to ONU O at an OLT (can be calculated from downstream broadcast policy such as round robin), then the weight W_{uv}^O on upstream link $(u,v) \in L_O$ is calculated as $P_{uv} + \sum_{\tilde{O} \neq O} D_{\tilde{O}}$ and the weight W_{uv}^O on downstream $(u,v) \in L_O$ is calculated as $P_{uv} + D_W$ [5].

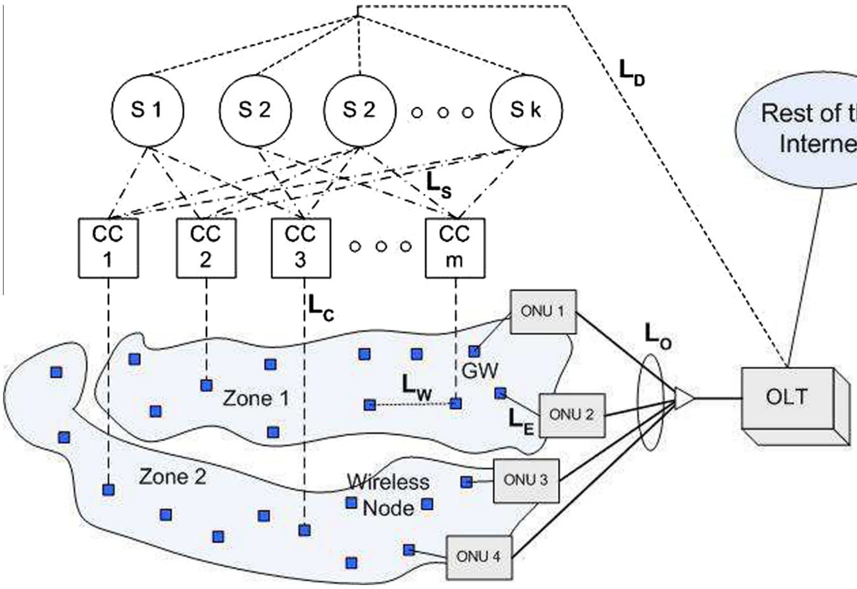

Fig. 4. Auxiliary graph for GRC algorithm.

5.1.4. Path computation for cloud traffic

GRC routes cloud traffic to any active CC hosting the requested service by calculating the serving delay on each CC. If the CC is active, the serving delay assigned is the weight W^C_{uv} on link $(u, v) \in L_C$ incident to it, otherwise $W^C_{uv} = \infty$. Considering independent arrivals, the serving delay for a CC is calculated as $\frac{1}{S_u - \frac{1}{\mu}[\sum_v(\lambda_{uv} - \sum_v \lambda_{vu}) - \gamma_u]}$, where S_u is the average service rate of a CC at node u, $\frac{1}{\mu}[\sum_v(\lambda_{uv} - \sum_v \lambda_{vu}) - \gamma_u]$ is the average request arrival rate [6], v is a neighbor of u, γ_u is the total traffic load to node u, and μ is the average request length. As each CC hosts and serve different services together, their aggregated arrival and service rate is considered.

For each link $(u, v) \in L_S$, a zero weight ($W^S_{uv} = 0$) is assigned. If a service S_k is hosted in CIW, weight on link $(u, v) \in L_D$, $W^{D,k}_{uv} = \epsilon$. As a result, in the auxiliary graph in Fig. 4, for a cloud service S_k, GRC performs anycast routing from any node u to the OLT and the request is served by any host CC of S_k that has lower serving delay. Here, the OLT is considered as the destination to perform anycast routing; the services are actually served from CCs. If a specific service is unavailable, then $W^{D,k}_{uv} = \infty$ and the service request goes through the OLT.

5.2. Energy savings in GRC

It has been observed in [13] that a WOBAN (and essentially a CIW) may serve different regions with different traffic profiles. We call each such region a “zone”. When the traffic demand is high in one zone, it may be low in another. For example, during the daytime, the downtown area of a city is busier, and may generate higher traffic, while in the evening the residential area may become busier. When the traffic load is lower, fewer ONUs are needed to carry the traffic and the rest can be turned off for green routing. They are switched on when the load gets higher. Note that the energy-saving aspect of GRC is not generally applicable to CIW-I because the CCs are integrated with the wireless nodes and it is not feasible to turn an ICC off if there is any wireless traffic at the host node. However, If a wireless node is inactive, it can go to sleep, which in turn will put the ICC to sleep.

5.2.1. Enabling technologies

Even though there is no power-saving mode defined for ONUs in IEEE 802.3 ah/802.3 av standards, there are proposals to include power-saving mode for ONUs [23]. This power-saving mode will put ONUs to sleep when they are idle. It is possible to attain up to 90% power savings by putting ONUs to sleep [13,23]. In a CIW, utilizing the centralized architecture of PON, the OLT can control the power savings of ONUs.

Similarly, CCs can be put to sleep when they have low load [9,10]. These devices can be remotely controlled, essentially by the OLT, and can be turned off when not needed. Because of deployment of each service over multiple CCs [33], it is not required to keep all the CCs in a zone active during a low-load period.

5.2.2. Energy saving strategy in GRC

Based on this idea, GRC utilizes the PON architecture for energy savings in CIW. As it is possible to do anycast routing over CIW, if V_Z is the set of active nodes in a zone Z, it is enough to keep x_Z ONUs active in zone Z such that $x_Z * C_{ONU} \geqslant \sum_{u \in V_Z} \gamma_u$, where C_{ONU} is the capacity of an ONU and γ_u is the demand at node u. Depending on the load, the OLT decides how many ONUs are needed for each zone.

This can be viewed as the OLT creating a "optical bandwidth pipe" to each zone with granularity equal to C_{ONU}. In the same way, for each zone Z, it is enough to keep y_Z CCs active such that $\left(y_Z * S_u \geqslant \sum_{u\in V_Z} \frac{1}{\mu}\left[\left(\sum_v \lambda_{uv} - \sum_v \lambda_{vu}\right) - \gamma_u\right]\right)$.

5.3. Wireless hop control

The energy-saving mechanism described in Section 5.2 may lead to an excessive number of wireless hops, which may degrade the performance. To avoid this, as shown in Fig. 5, GRC deploys a wireless hop control strategy. GRC creates a breadth-first search tree (BFST), T_O, rooted at each ONU O, that reaches each wireless node in CIW within h_O hops. In other words, each ONU O creates a BFST of a predetermined length = h_O. While selecting the x_Z ONUs for each zone, the OLT ensures that all the active nodes are within at least one T_O, where O is an active ONU. As each T_O is *pre-computed*, selecting an active ONU requires just tree traversals of T_O at each ONU and determining if all the active nodes are at least on one of the trees. This ensures all the wireless nodes in zone Z are within h_O hops of all the active ONUs. For example, in Fig. 5, both ONU 1 and ONU 2 create BFSTs for $h_O = 2$ and by checking these two BFSTs, GRC can determine if a node is covered by an ONU within h_O wireless hops. Similarly, a BFST, T_C, is computed at each CC with a maximum predetermined length = h_C. OLT selects y_Z active CCs in zone Z such that each node $u \in V_Z$ can find at least one host CC for each of its requested cloud services within h_C hops by traversing T_Cs.

5.4. Self-management and adaptability of GRC

One of the most prominent features of GRC is that it can self-manage to recover from failures and adapt itself to various scenarios. Here, we discuss these features.

- *Failure recovery:* As GRC is designed to operate with turned-off network components, it can self-manage itself for failure recovery. It receives failure information from LSAs and it adjusts the traffic flow accordingly. In case of an ONU or a CC failure, GRC simply considers it to be turned-off and assigns $W^E_{uv} = \infty$ or $W^C_{uv} = \infty$, respectively. In the same way, if a wireless node fails, GRC considers it to be inactive, and assigns $W^W_{uv} = \infty$ for all the links incident to the failed node.
- *Adaptability:* GRC can be adapted to operate over WOBAN, or any other network architecture with a wireless front-end and a centralized backhaul. For example, if GRC is to deployed over WOBAN alone, all it has to do is to consider no service is deployed over WOBAN and assign $W^C_{uv} = \infty$ as described in Section 5.1.4. Then, none of the traffic will be routed to CCs and it would perform power-saving, anycast routing for only the regular traffic.

  Another important aspect of GRC is that, although power savings is a key feature of the algorithm, GRC is not dependant on it. If green routing is not required, it can operate as an efficient routing algorithm for CIW and WOBAN by performing anycast routing over shortest-delay paths for both regular and cloud traffic without turning off any network device. This can be done by assigning the value of x_Z and y_Z to be all ONUs and CCs at each zone, respectively. Node that, without the power-saving feature, GRC performs with similar efficiency as CaDAR [5], which is shown to be an efficient routing algorithm for WOBAN, because they operate on same principles.

Algorithm 1 describes the details of GRC.

Algorithm 1. GRC Algorithm

```
Precomputation:
    (i) Compute BFST, $T_O$, over $\{L_W \bigcup L_E\}$ rooted from each ONU $O$ with length = $h_O$
    (ii) Compute BFST, $T_C$ over $\{L_S \bigcup L_C \bigcup L_W\}$ rooted from each CC $C$ with length = $h_C$
1. Link-State Advertisement (LSA): For each link $(u, v)$ from node $u$, advertise periodically the capacity ($C_{uv}$), flow ($\lambda_{uv}$), and time stamp to wireless nodes
2. Link-Weight Assignment:
    (i) $W^W_{uv} = \frac{1}{C_{uv}-\lambda_{uv}} + \frac{1}{2C_{uv}}$
    (ii) $W^E_{uv} = \frac{1}{C_{uv}-\lambda_{uv}}$ if active, $W^E_{uv} = \infty$ otherwise
    (iii) $W^O_{uv} = P_{uv} + \sum_{\tilde{O}\neq O} D_{\tilde{O}}$ for upstream, $W^O_{uv} = P_{uv} + D_W$ for downstream
    (iv) $W^C_{uv} = \frac{1}{S_u - \frac{1}{\mu}\left[\sum_v\left(\lambda_{uv} - \sum_v \lambda_{vu}\right) - \gamma_u\right]}$ if active, $W^C_{uv} = \infty$ otherwise
    (v) $W^S_{uv} = 0$
    (vi) $W^{D,k}_{uv} = \epsilon$ if service is available, $W^{D,k}_{uv} = \infty$ otherwise
3. Path Computation:
    (i) For regular traffic, compute the shortest-delay path over $\{L_W \bigcup L_E \bigcup L_O\}$
    (ii) For cloud traffic, compute the shortest-delay path over $\{L_W \bigcup L_S \bigcup L_C\}$ if service is available, over $\{L_W \bigcup L_E \bigcup L_O \bigcup L_D\}$ otherwise
4. Active ONU Selection: For each zone $Z$, select $x_Z$ active ONUs such that $(x_Z * C_{ONU} \geqslant \sum_{u\in V_Z} \gamma_u)$ and each node $u \in V_Z$ is within $h_O$ hops of an active ONU by traversing $T_O$
5. Active CC Selection: For each zone $Z$, select $y_Z$ active CCs such that $\left(y_Z * S_u \geqslant \sum_{u\in V_Z} \frac{1}{\mu}\left[\left(\sum_v \lambda_{uv} - \sum_v \lambda_{vu}\right) - \gamma_u\right]\right)$ and each node $u \in V_Z$ can find at least one host CC for each of its requested cloud services within $h_C$ hops by traversing $T_C$
```

6. Illustrative numerical examples

To study the performance of a CIW, we use a 164-node hypothetical WOBAN over Davis, CA, as shown in Fig. 6 with 12 ONUs and 2 GWs per ONU serving three zones (residential, university campus, and downtown). Each radio has a capacity of 54 Mbps. We analyze the impact of different deployment of services in WOBAN and how it affects the performance. We assume capacity allocation over wireless

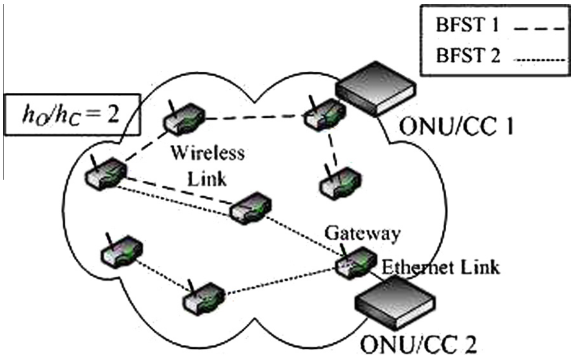

Fig. 5. Application of BFST to enforce wireless hop control in GRC.

links is done by TDM scheduling. ONU and OLT have capacities of 100 Mbps and 1 Gbps, respectively.

The MILP in Section 4 gives the design of a CIW by providing the placements of CCs and the services they host. After obtaining the design, we highlight the operational aspect using GRC and obtain the flows on links for our performance study. First, in Section 6.1, we evaluate the merits of CIW and use GRC without turning off any ONU or CC. Then, in Section 6.2, we analyze the performance of the GRC algorithm by evaluating its energy-saving mechanism and the corresponding system delay.

6.1. Performance evaluation: CIW design

We evaluated the performance of both CIW-S and CIW-I described in Section 3.2. We solved the MILP described in Section 4 using ILOG CPLEX 9.0 on an Ubuntu Linux operating system on an Intel Core 2 Duo machine with 4 GB of RAM and design the CIW by distributing 8 and 12 cloud services to CCs and assign the CCs to different nodes accordingly with a restriction of maximum wireless hops, h_C, as 2 and 3. Processing and memory requirements for each service are generated randomly and they vary across services. We assume that a CC can host several services, based on the processing and memory requirements. We created the demand matrix of services by taking the average of 1000 randomly-generated demands for each service for each node. Depending on which zone a node is in, it will have different demand for different services.

In our evaluation, we first show a design of a CIW obtained from the MILP in Section 4. Then, we use GRC to show the performance of our design for different input parameters. We use *system delay*, which is the average network-wide packet delay [34], as the performance metric to compare the performance of various implementations of CIW for the network in Fig. 6.

For fair comparison, we first insert the cloud traffic in the network. The average load of cloud traffic for 8 services is about 2.5 Mbps at each node. Then, we increase the

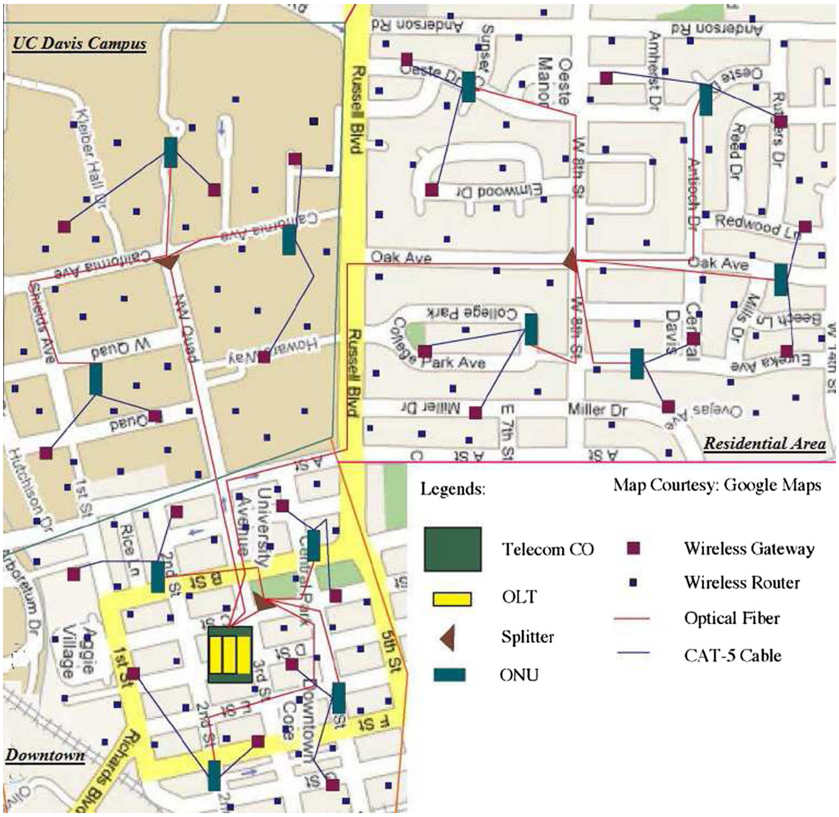

Fig. 6. Topology for analytical study.

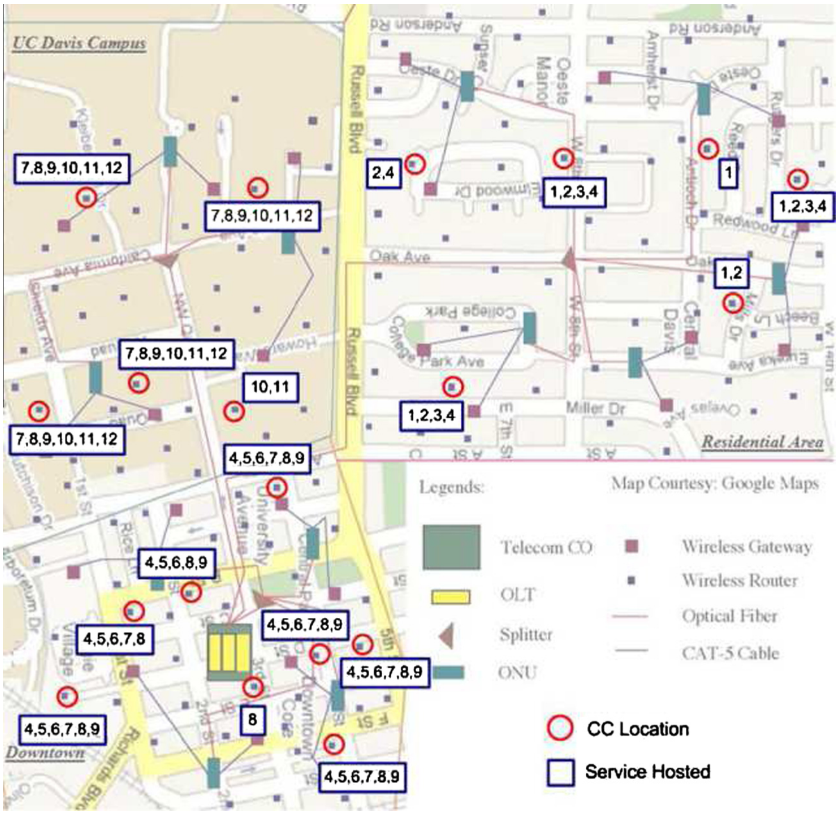

Fig. 7. Deployment of CCs and services for $K = 12$ and $h_C = 2$.

background traffic and evaluate the performance. Service traffic that are not served locally are included in the background traffic. In this way, we obtain the overall system delay based on the total (both cloud and background) traffic demand. Note that Figs. 8–12 have the same cloud traffic; so we only compare based on the additional background traffic.

Fig. 7 shows the deployment of SCCs and their service assignment for the topology in Fig. 6. We observe that 12 services can be deployed with CCs deployed at only about 12% of wireless nodes in the topology. We also observe that some of the CCs host more services while some host as few as one. This is because of the effect of random service demand and the restriction of a maximum 2 wireless hops

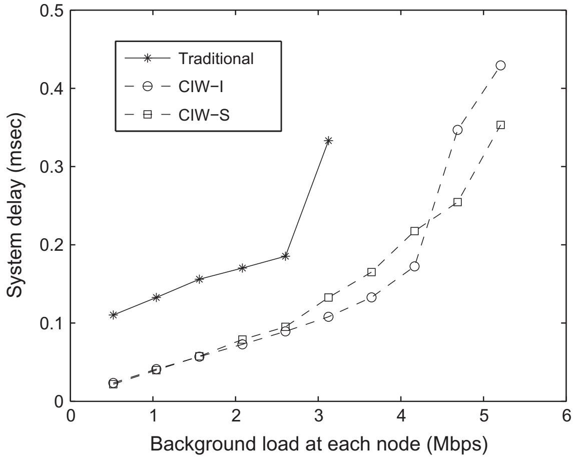

Fig. 8. System delay vs. background load for $K = 8$ and $h_C = 2$.

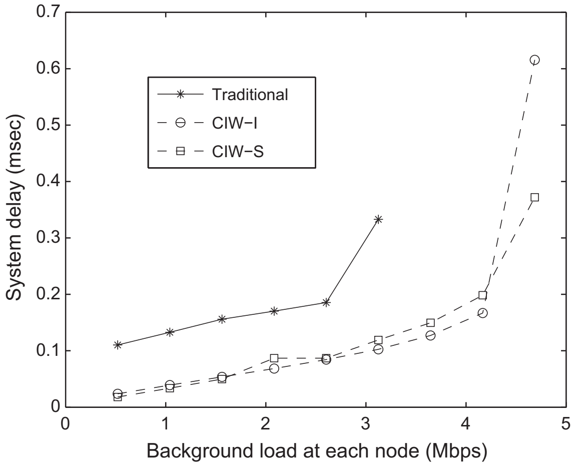

Fig. 9. System delay vs. background load for $K = 12$ and $h_C = 2$.

(h_C) taken by service traffic. It also shows that the processing and memory capability of SCCs, as highlighted in Eqs. (4) and (5), can vary depending on the number and types of services they are hosting.

Fig. 8 shows the impact of deploying eight services ($K = 8$) for a CIW for the topology in Fig. 6. Here, service delivery from the cloud is allowed to travel two wireless hops ($h_C = 2$). We see that CIW carries almost twice the background traffic after satisfying the cloud's demands with about one-third of the system delay than a traditional deployment of WOBAN. This performance improvement is due to offloading of cloud traffic from wireless backhaul. We see that CIW-S carries the same traffic as the CIW-I with slightly higher delay. This is because the CIW-S minimizes the number of deployed SCCs, and hence introduces some traffic in the WMN while nodes with CIW-I mostly serve locally. Similarly, Fig. 9 shows the performance of a CIW for $K = 12$ and $h_C = 2$. We observe that, for $K = 12$, CIW carries the same background traffic as $K = 8$ with slightly lower system delay as more cloud services are served within the network.

For sensitivity analysis, Fig. 10 shows the system delay variations for increasing cloud traffic for $h_C = 2$ for 12 services. We have varied the service traffic load up to three times to observe the impact of service traffic load on the performance of CIW. We observe that, even with lower service traffic demand, the performance of our design is better than the traditional approach. With increasing service load, our approach performs consistently better. We observe that, as expected, higher service traffic leads to higher system delay.

Fig. 11 shows the performance of a CIW for $K = 8$ and $h_C = 2$ and 3 (4 or more hops carries traffic to gateways from any node in Fig. 6; moreover, more wireless hops will result in degraded performance). We observe that increasing the number of hops does not impact the performance of CIW-I as most of the services are served locally by this approach. For CIW-S with 3 hops, the number of SCC deployed in the network decreases (Fig. 13) as cloud traffic is allowed to travel more hops to satisfy the demands. But

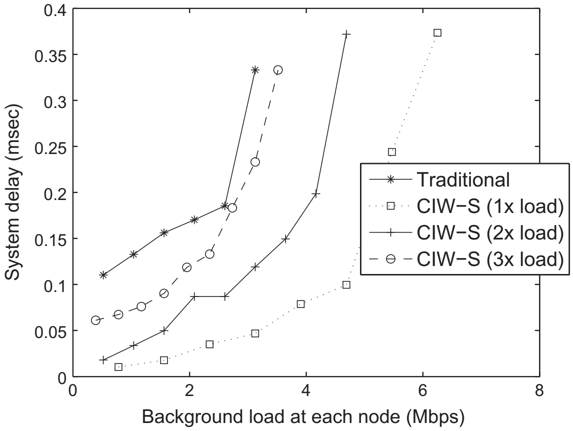

Fig. 10. System delay vs. background load for $K = 12$ and $h_C = 2$ for sensitivity analysis.

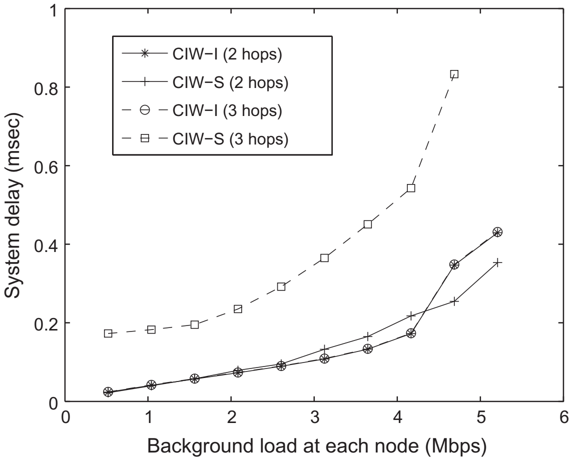

Fig. 11. System delay vs. background load for $K = 8$ and $h_C = 2$ and 3.

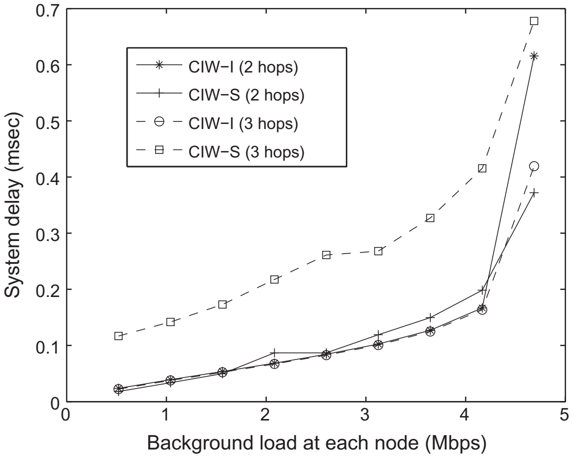

Fig. 12. System delay vs. background load for $K = 12$ and $h_C = 2$ and 3.

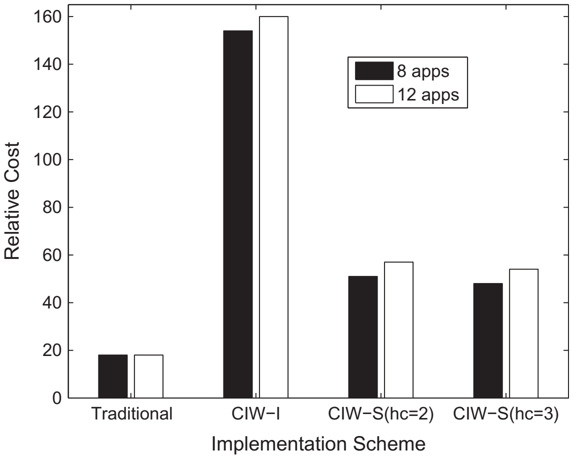

Fig. 13. Relative cost of different implementation models.

this leads to higher system delay and reduced background traffic carried by CIW.

Fig. 12 shows the performance of the CIW when the number of cloud services served by the network is 12. We observe that CIW-I carries higher background traffic as most of the service traffic is absorbed locally. We see that CIW-S with 3 hops has higher delay than 2 hops because it introduces more traffic to the network. But it carries more background traffic as it diverts more traffic away from the gateways.

Fig. 13 shows the relative cost of traditional WOBAN, CIW-I, and CIW-S for the topology in Fig. 6. We consider the cost of wireless routers as a unit cost. Because SCCs are built on system boards [10] and require additional memory, we assume that an SCC costs twice as much as a wireless router [10]. We assume one cloud server is required to host the services for traditional implementation, though this number may vary depending on deployment policy. We consider that the cost of a server is twice than that of an SCC. Because of volume discount, we consider the cost of ICC to be 2.5 times the cost of a wireless node. We see from Fig. 13 that CIW-S requires a slight increase of cost while CIW-I costs almost twice as much as the traditional approach. This is because one SCC can host several services and the objective of our placement formulation (Eq. (1)) for CIW-S minimizes the number of SCC needed to host all the services. But for CIW-I, only a few services can be hosted at each ICC and each node may require an ICC to host all the services.

We see from our performance evaluation that CIW carries higher traffic from the wireless front-end of WOBAN with lower system delay. Among the deployment options of CIW, CIW-I achieves better performance, but does not support more services, and has higher cost compared to CIW-S. On the other hand, CIW-S achieves significant performance gain over the traditional implementation with a slight increase of cost, and can support higher number of services compared to CIW-I.

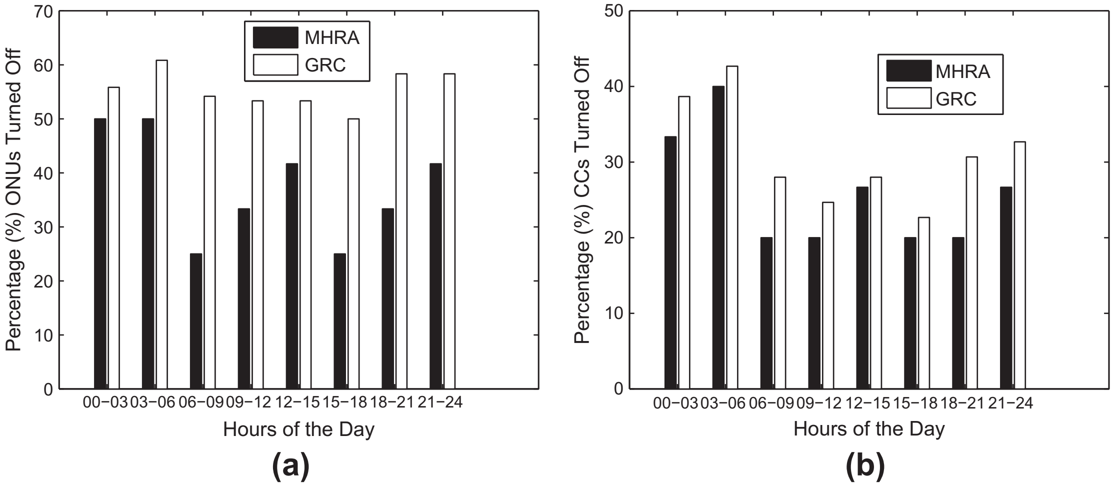

Fig. 14. Power savings with ONU (a) and power savings with CC (b).

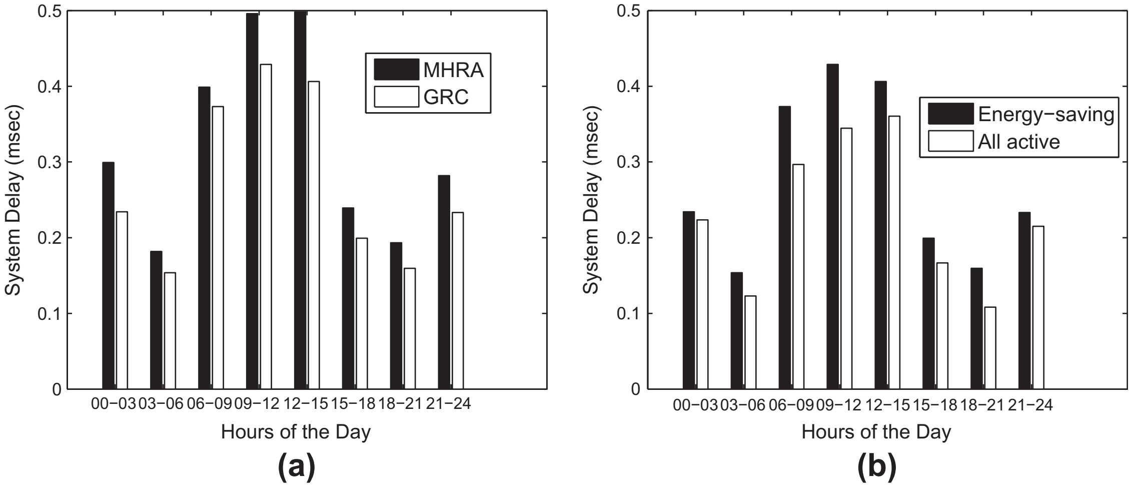

Fig. 15. System delay (MHRA vs. GRC) (a) and impact of energy savings on system delay (b).

6.2. Performance evaluation: energy savings using GRC

To study the effectiveness of GRC for CIW, we deploy 6, 5, and 4 SCCs at each of the zones to host a total of 12 cloud services with different memory and processing requirements. We consider h_O and h_C to be 3. In our illustrative examples, performance evaluations are averaged over 100 simulation runs with randomly-selected active wireless nodes in each zone. We also use the same daily traffic profile for each zone as in [13] where the 24 h of a day are divided into eight periods. We consider that each zone has different types of users with different behavior patterns and different usage rate. A combination of these two factors determines the traffic profile [13]. For each simulation run, the fraction of service traffic to the total traffic is randomly selected from a uniform distribution between 0.1 and 0.7. We assumed that there will be at least 30% background traffic for each node. We compare the performance of GRC versus a Minimum-Hop Routing Algorithm (MHRA). In order to turn off ONUs and CCs, we apply a *baseline greening technique* that turns off an ONU or CC if its load is below a certain threshold, as in [13]. For our performance evaluation, we used a threshold of 10%. We assume capacity allocation over wireless links is done by TDM scheduling. ONU and OLT capacities are 100 Mbps and 1 Gbps, respectively.

Fig. 14(a) shows the energy savings in terms of the percentage of ONUs turned off during different periods of the day. We see that, on an average, about 50% of the ONUs can be turned off using either of the schemes for each algorithm. However, depending on the time of the day, GRC turns off 5–20% more ONUs than MHRA and as a result, saves more energy.

Similarly in Fig. 14(b), we show that about 20% of the CCs can be turned off by either of the schemes though GRC achieves about 3–10% additional energy savings. It is important to note that, in our setting, even though 50% of the ONUs and about 20% of the CCs can be turned off in low-load periods, all of them are still needed during high-load periods (when none of the ONUs and CCs in a particular zone may get turned off).

Fig. 15(a) shows the system delay for different periods of the day. We observe that GRC has about 2–7% lower delay than MHRA depending on the period of the day. This is because MHRA packs traffic on links to divert load to the currently-active ONUs and CCs, and turns them off depending on their current load. This usually leads to uneven distribution of traffic and higher system delay. On the other hand, GRC keeps ONUs and CCs active considering the load on each zone and performs anycast routing among the active devices. This approach not only provides energy savings but also distributes the load across the network, leading to lower delay.

Fig. 15(b) shows the impact of energy savings on the average packet delay of GRC. Here, for the "all active" option, we keep all the ONUs and CCs active for each period. We observe that, when GRC performs in power-saving mode, it increases the system delay by about 5–10% depending on the period of the day. The slight degradation of performance is expected as GRC shuts down about half the network components to save power. We also observe that, even with all the ONUs and CCs active, the system delay does not decrease significantly. This is a result of strategically turning off network components and intelligent, load-balanced routing among active components by GRC. Fig. 15(b) also shows that GRC can operate without power saving as well.

We see from our performance evaluation that GRC achieves better power savings consistently from both CCs and ONUs compared to MHRA. We also observe that GRC, due to its efficient packing and distribution of traffic, achieves lower system delay compared to MHRA. GRC further reduces the system delay if it operates to keep all CCs and ONUs active instead of operating in a power-saving mode.

7. Conclusion

Service-centric behavior of end users is shaping the traffic pattern of today's networks. In a WOBAN, accessing a cloud takes up wireless backhaul bandwidth, creates bottleneck at the gateways, and has a possibility of stale server updates. Equipping WOBAN with cloud components creates an integrated platform for cloud traffic, called CIW, that addresses these limitations. We presented a design scheme for CIW using an MILP. We also proposed a scheme, GRC, to self-manage network component and traffic over CIW. GRC saves power by selectively turning off networking devices when possible while making sure the active devices can serve the traffic demands of the network. It distributes the load across the network by performing anycast routing among the active devices. Though designed for CIW, GRC can also operate on WOBAN alone and can route traffic with or without power-saving mode. We observed from our performance study that CIW can achieve significant performance improvement through offloading of traffic even for low service demands. We also observed that GRC achieves similar power savings with lower delay than baseline greening techniques because of its routing strategy. We conclude that CIW is a technologically-viable access network design that can improve the performance of a WOBAN in a cost-effective way and GRC is an adaptable, power-saving routing algorithm for CIW with improved performance.

Acknowledgment

A short, summarized, and partial version of this paper has been presented at the IEEE International Conference on Communications (ICC) in Kyoto, Japan, in June 2011 [33].

References

[1] Butler Group, Application Delivery: Creating a Flexible, Service-Centric Network Architecture, September 2007. <http://www.mindbranch.com/Application-Delivery-Creating-R663-21/>.

[2] S. Sarkar, S. Dixit, B. Mukherjee, Hybrid wireless-optical broadband access network (WOBAN): a review of relevant challenges, IEEE/OSA J. Lightwave Technol. 25 (11) (2007) 3329–3340.

[3] Strix Systems, Commercial Deployments. <http://www.strixsystems.com/newsspotlight.aspx>.

[4] G. Kramer, Ethernet Passive Optical Networks, McGraw-Hill Professional, 2005.

[5] A. Reaz, V. Ramamurthi, S. Sarkar, D. Ghosal, S. Dixit, B. Mukherjee, CaDAR: an efficient routing algorithm for a wireless-optical broadband access network (WOBAN), J. Opt. Commun. Netw. 1 (5) (2009) 392–403.

[6] S. Sarkar, H.H. Yen, S. Dixit, B. Mukherjee, A novel delay-aware routing algorithm (DARA) for a hybrid wireless-optical broadband access network (WOBAN), IEEE Netw. 22 (3) (2008) 20–28.

[7] Emerson Process Management, Wireless Field Data Backhaul, Service Data Sheet, October 2012.

[8] A. Reaz, V. Ramamurthi, M. Tornatore, S. Sarkar, D. Ghosal, B. Mukherjee, Cost-efficient design for higher capacity hybrid wireless-optical broadband access network (WOBAN), Comput. Netw. 55 (9) (2011) 2138–2149.

[9] CTERA Networks, Cloudplug. <http://www.ctera.com/home/ctera-cloudplug.html>.

[10] PC Engines, ALIX System Boards. <http://www.pcengines.ch/alix.htm>.

[11] L. Srinivasan, J. Treadwell, An overview of service-oriented architecture, web services and grid computing, HP Softw. Glob. Bus. Unit 2 (2005).

[12] P. Donegan, R. Brandon, A. Jones, T. Naveh, T. Hack, Backhaul to the future: mobile broadband profitability requires smarter backhaul networks. Light Reading Webinar, September 2010.

[13] P. Chowdhury, M. Tornatore, S. Sarkar, B. Mukherjee, Building a green wireless-optical broadband access network (WOBAN), IEEE/OSA J. Lightwave Technol. 28 (16) (2010) 2219–2229.

[14] T. Erl, Service-Oriented Architecture: A Field Guide to Integrating XML and Web Services, Prentice Hall PTR, 2004.

[15] N. Bieberstein, S. Bose, L. Walker, A. Lynch, Impact of service-oriented architecture on enterprise systems, organizational structures, and individuals, IBM Syst. J. 44 (4) (2005) 691–708 (No. 4).

[16] Zeus Technology Limited, Building a Service Oriented Network Using a Service Delivery Controller, 2007. <http://www.zeus.com/documents/en/Bu/Building_a_Service_Oriented_Network.pdf>.

[17] D. Gračanin, M. Eltoweissy, A. Wadaa, L. DaSilva, A service-centric model for wireless sensor networks, IEEE J. Sel. Areas Commun. 23 (6) (2005) 1159–1166.

[18] B. Rao, L. Minakakis, Evolution of mobile location-based services, Commun. ACM 46 (12) (2003) 61–65.

[19] J. Leblet, Z. Li, G. Simon, D. Yuan, Optimal network locality in distributed virtualized data-centers, Comput. Commun. 34 (16) (2011) 1968–1979.

[20] N. Laoutaris, G. Smaragdakis, K. Oikonomou, I. Stavrakakis, A. Bestavros, Distributed placement of service facilities in large-scale networks, in: Proc. IEEE INFOCOM, Anchorage, AK, 2007, pp. 2144–2152.

[21] G. Pallis, A. Vakali, Insight and perspectives for content delivery networks, Commun. ACM 49 (1) (2006) 101–106.

[22] V. Valancius, N. Laoutaris, L. Massoulie, C. Diot, P. Rodriguez, Greening the internet with nano data centers, in: Proc. International Conference On Emerging Networking Experiments And Technologies, Rome, Italy, December 2009, pp. 37–48.

[23] J. Mandin, EPON Power Saving Via Sleep Mode, in: IEEE P802.3av 10G EPON Task Force Meeting, September 2008.

[24] J. Zhang, N. Ansari, Toward energy-efficient 1g-EPON and 10g-EPON with sleep-aware MAC control and scheduling, IEEE Commun. Mag. 49 (2) (2011) s33–s38.

[25] L. Lin, N. Shroff, R. Srikant, Asymptotically optimal energy-aware routing for multihop wireless networks with renewable energy sources, IEEE/ACM Trans. Netw. 15 (5) (2007) 1021–1034.

[26] A. Mohanoor, S. Radhakrishnan, V. Sarangan, Online energy aware routing in wireless networks, Ad Hoc Netw. 7 (5) (2009) 918–931.

[27] D. Yang, X. Li, R. Sawhney, X. Wang, Geographic and energy-aware routing in wireless sensor networks, Int. J. Ad Hoc Ubiquitous Comput. 4 (2) (2009) 61–70.

[28] V. Ramamurthi, A. Reaz, B. Mukherjee, Optimal capacity allocation in wireless mesh networks, in: Proc. IEEE Globecom, New Orleans, LA, December 2008.

[29] Devcom Solutions AB, Wisper Wireless. <http://www.wisper.se/>.

[30] Firetide, Wireless Mesh Products. <http://www.firetide.com/>.

[31] Tropos Networks, Wireless Mesh Products. <http://www.tropos.com/products/index.html>.

[32] Meraki, Wireless Mesh Products. <http://meraki.com/products_services/hardware/>.

[33] A. Reaz, V. Ramamurthi, M. Tornatore, Cloud-over-WOBAN (CoW): an offloading-enabled access network design, in: Proc. IEEE International Conference on Communications (ICC), Kyoto, Japan, June 2011.

[34] L. Kleinrock, Queueing Systems, Volume II: Computer Applications, Wiley-Interscience, 1976.