Routing in Accumulative Multi-hop Networks

Jesús Gómez-Vilardebó, *Member, IEEE*

***Abstract*—This paper investigates the problem of finding optimal paths in single-source single-destination accumulative multi-hop networks. We consider a single source that communicates to a single destination assisted by several relays through multiple-hops. At each hop, only one node transmits, while all the other nodes receive the transmitted signal, and store it after processing/decoding and mixing it with the signals received in previous hops. This is, we consider that terminals make use of advanced energy accumulation transmission/reception techniques such us maximal ratio combining reception of repetition codes, or information accumulation with rateless codes. Accumulative techniques increase communication reliability, reduce energy consumption, and decrease latency. We investigate the properties that a routing metric must satisfy in these accumulative networks to guarantee that optimal paths can be computed with Dijkstra's algorithm. We model the problem of routing in accumulative multi-hop networks, as the problem of routing in a hypergraph. We show that optimality properties in traditional multi-hop network (monotonicity and isotonicity) are no longer useful and derive a new set of sufficient conditions for optimality. We illustrate these results by studying the minimum energy routing problem in static accumulative multi-hop networks for different forwarding strategies at relays.**

***Index Terms*—Accumulative multi-hop, energy accumulation, minimum energy, graph theory.**

I. Introduction

Introducing relay capabilities in a network has a strong effect on the information flow that extends to all communication levels, from the achievable rates to the routing strategy. A fundamental understanding of the role that relays play in wireless networks is of paramount importance for the design of efficient protocols in future communication systems.

The problem of routing in *traditional multi-hop* (TM) communication networks, where each relay node only listens to the immediately previous node is quite well understood today. For the purpose of routing, these networks are well modeled by directed graphs. Given a routing metric criteria, the optimality conditions that guarantee that efficient path search algorithms, such as Dijkstra's algorithm, find the optimal path were studied in [1], [2].

The problem of routing in *accumulative multi-hop* (AM) communication networks, in which we are instead interested here, is however far from being understood today. In the simplest accumulative multi-hop network, a single source communicates to a single destination assisted by several relay nodes that can accumulate the received energy/information from previous relay transmissions. In practice, there are two main accumulation mechanisms at relays: energy and mutual-information accumulation. Energy accumulation can be performed at the receiving nodes, e.g., through space-time coding or repetition coding [3], [4]. Mutual-information accumulation [5], [6] can be realized using rateless codes e.g. fountain or raptor codes [7]. Accumulation mechanisms are considered in current and next generation standards since they increase communication reliability and reduce energy consumption.

The work presented here builds, mainly, on top of the works conducted in [1], [2]. We show that the AM network communication routing problem can not be represented using graphs, and thus, the optimality conditions derived in [1], [2] for routing over graphs can not be invoked. We instead show that, in general, the AM routing problem needs to be modeled using hypergraphs. We then find new conditions to guarantee the optimality of Dijkstra's algorithm in hypergraphs. These conditions are only sufficient but not necessary.

Equipped with these optimality conditions, we discuss the optimality of Dijkstra's algorithm for the minimum energy routing problem in static AM networks. To that end, we focus mainly on decode-and-forward (DF) based relaying strategies. DF relay nodes decode the source message completely by accumulating energy, or information from all previous transmissions. This routing problem has been previously addressed in [3], [4], [7]–[10]. From [3] and [4], we already know that finding the optimal transmission order for these networks is an NP-complete problem.

Our approach here consist instead on identifying particular DF AM network situations for which the routing problem can be represented either using graphs that satisfy Dijkstra's optimality conditions in [2], or using hypergraphs that satisfy the new optimality conditions found here.

First, motivated by the observation that the TM network can be seen as a particular DF AM network where relays can only accumulate energy/information from the intermediately previous relay, we identify two other DF AM networks for which the routing problem can be also represented using a graph. We refer to them as the DF Source Accumulative Multi-hop (DF SAM) and the DF Destination Accumulative Multi-hop (DF DAM) networks. In the DF SAM network, relays and destination nodes decode the source message by accumulating the energy/information coming from the immediately previous relay and the source. Instead, in the DAM network relay nodes decode the source message by accumulating the energy/information coming from the immediately previous relay, whereas the destination accumulates the energy/information from every node in the path. We then consider the cut-set bound (CB) for AM networks [11, Th. 14.10.1]. The CB outperforms any possible forwarding strategy. For the CB, we show that the routing problem is modeled by a hypergraph and satisfies the new sufficient condition for optimality.

This work was partially supported by the Catalan Government under grant SGR2014-1567, the Spanish Government under grant PCIN-2013-027 (E-CROPS), and TEC2013-44591-P (INTENSYV), and the European project FP7-ICT-2011-8 NEWCOM# 318306.

J. Gómez-Vilardebó was with the Centre Tecnològic de Telecomunicacions de Catalunya (CTTC/CERCA), Barcelona 08860, Spain (e-mail: jesus.gomez@cttc.cat)

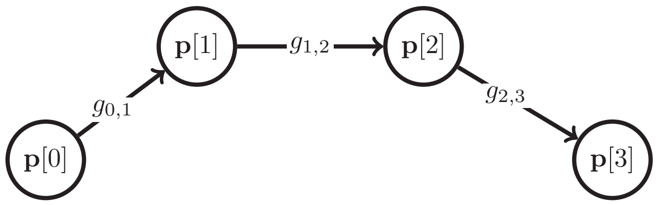

Fig. 1. TM communication model.

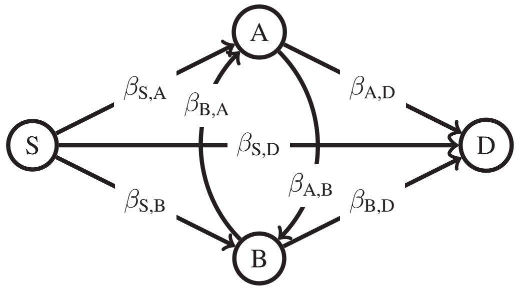

Fig. 2. TM directed graph model ($N = 4$) for unicast multi-hop routing in wireless communication channels.

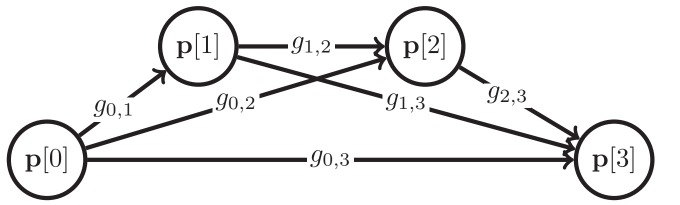

Fig. 3. AM communication model.

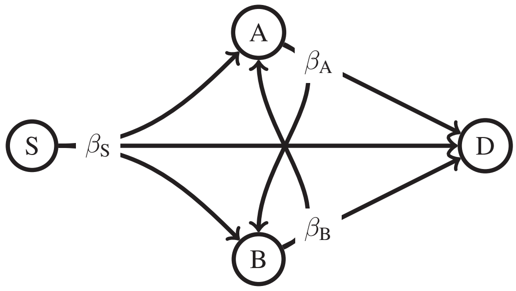

Fig. 4. AM hypergraph model ($N = 4$) for unicast accumulative multi-hop routing in wireless communication channels. If we associate one metric for each of the vertices of the hyperedge, then $\beta_{\text{S}} = \{\beta_{\text{S,A}}, \beta_{\text{S,B}}, \beta_{\text{S,D}}\}$, $\beta_{\text{A}} = \{\beta_{\text{A,B}}, \beta_{\text{A,D}}\}$, $\beta_{\text{B}} = \{\beta_{\text{B,A}}, \beta_{\text{B,D}}\}$.

Motivated by this success, we show that for the DF AM network there exist a subset of paths also satisfying the new optimality conditions. In that case, we are able to keep all the energy/information accumulation at nodes at the cost of possibly finding a suboptimal path. Finally, besides the DF relaying, we also consider the parity-forwarding (PF) strategy introduced in [7] applied to the DAM network. We show that in the PF DAM network, the routing problem can be represented using a graph and satisfies Dijkstra's optimality conditions.

We have presented the Dijkstra's optimality conditions for hypergraphs previously in [12]. Here, we provide a more detailed derivation. There, we also studied the minimum energy routing problem for the TM, DF AM and CB networks. However, the DF SAM, DF DAM, PF DAM and DF EAM networks are first discussed here. Finally, while this work deals mostly with optimal path search algorithms for AM networks, we propose in [13] good heuristic path search algorithms for the DF AM network.

The remainder of the paper is organized as follows. The AM network model is presented in Section II. In Section III, the minimum energy accumulative path weight function for DF, PF relaying and for the CB are derived. The optimality of Dijkstra' algorithm in AM networks is discussed in Section IV, and particularized for the minimum energy accumulative routing metrics in Section V. Finally, conclusions are drawn in Section VII.

II. The Accumulative Network Model

Consider a static network with N nodes. The traffic is unicast, from a source node (S) to a destination node (D) with the help of relay transmissions. Relay nodes transmit according to a given transmission order which is described by a path vector $\mathbf{p}$, where $\mathbf{p}[0] = \text{S}$, $\mathbf{p}[L+1] = \text{D}$, and $L \leq N-2$ is the number of relays. Notice that we only allow one relay node in each path position. Communications can be either point-to-multipoint as in wireless channels, or point-to-point as in wireline channels.

In TM communications, see Fig. 1, given a path $\mathbf{p}$, the signal transmitted by node $\mathbf{p}[i]$ is only intended to node $\mathbf{p}[i+1]$. This is so, even if transmissions are over wireless channels, and the transmitted signals are also overhead by nodes in the path other than the intended ones. These nodes ignore or treat as interference the non intended received signals. In TM routing problems, the network is well modeled by a directed graph $\mathcal{G}(V, E)$, as the one shown in Fig. 2, where V is the set of nodes and E is the set of edges representing the existence of links between pairs of nodes. Let $e_{\text{u,v}}$ denote the edge between nodes u and v. A path $\mathbf{p}$ exits if $e_{\mathbf{p}[i],\mathbf{p}[i+1]} \in E$ for all $i = \{0, ..., L\}$. Associated to each edge, there can be one or several fixed metrics, e.g. the link distance, the link bandwidth, the channel magnitude, the transmission delay, etc. For simplicity, let us as assume that there is only one metric per edge, then $\beta\left(e_{\text{u,v}}\right) = \beta_{\text{u,v}}$ denotes the metric associated to edge $e_{\text{u,v}}$. In Fig. 2, we depict a TM graph model for routing in wireless communications from a source node S to a destination node D using two possible relay nodes A and B. Observe that the source node can only transmit to every other node but can not receive from them. Contrary, the destination can only receive from every other node. Relays can transmit to every other node (except the source) and receive from every other node (except destination). In TM routing the objective is to find the better path, or *lightest path*, between a source and a destination according to some network metric. The weight of a path $w(\mathbf{p})$ is a function of the metrics of the edges traversed by a path, namely $w(\mathbf{p}) = w\left(\beta_{\mathbf{p}}\right)$, where $\beta_{\mathbf{p}} = \left\{\beta_{\mathbf{p}[i],\mathbf{p}[i+1]}, i = 0, ..., L\right\}$. A commonly used path weight function in TM routing is the summation of the link weights of the edges traversed by the path

$$w(\mathbf{p}) = \sum_{i=0}^{L} w(e_{\mathbf{p}[i],\mathbf{p}[i+1]}). \qquad (1)$$

This path weight function admits the following recursive expression from the weight of the partial paths $\mathbf{p}_i =$

$\langle \mathbf{p}[0], \mathbf{p}[1], ..., \mathbf{p}[i] \rangle$

$$w(\mathbf{p}_i) = w\left(e_{\mathbf{p}[i],\mathbf{p}[i+1]}\right) + w(\mathbf{p}_{i-1}),$$

for $i = 1$ to i= $L+1$, then $w(\mathbf{p}) = w(\mathbf{p}_{L+1})$.

As an example, if we consider the channel gain between node node u and v, $g_{\mathrm{u,v}}$, as the metric associated to the edge $e_{\mathbf{u},\mathbf{v}}$, namely $\beta_{\mathbf{u},\mathbf{v}} = g_{\mathrm{u,v}}$, and define the weight of an edge as $w\left(e_{\mathbf{u},\mathbf{v}}\right) = \frac{1}{\beta_{\mathbf{u},\mathbf{v}}}$ then (1) equals $w(\mathbf{p}) = \sum_{i=0}^{L} \frac{1}{g_{\mathbf{p}[i],\mathbf{p}[i+1]}}$ which, as we show in Section III, for a particular communication model measures the energy efficiency of the path $\mathbf{p}$ in TM networks.

Finally, observe that the path weight function

$$w(\mathbf{p}_i) = \beta_{\mathbf{p}[i-1],\mathbf{p}[i]} + \beta_{\mathbf{p}[i-2],\mathbf{p}[i]} + w(\mathbf{p}_{i-1}),$$

can not be computed over a graph as the edge $e_{\mathbf{p}[i-2],\mathbf{p}[i]}$ although it may exist it is not traversed by the path $\mathbf{p}$. We will encounter this type of path weight functions studying AM communication networks next.

In AM communications, see Fig. 3, in which we are instead interested, relay nodes do not discard the received signals from previous nodes in the path. This is, relay and destination nodes may benefit from the signals received from all previous nodes in the path. In AM routing, the network is better modeled by a directed hypergraph $\mathcal{H}(V, E)$, as the one shown in Fig. 4, where V denotes the set of nodes, or vertices, and E denotes the set of hyperedges, or connections between nodes. A directed hypergraph is a generalization of a directed graph in which each hyperedge is allowed to have multiple source (tail) vertices and multiple destination (head) vertices. The tail and head vertices of a hyperedge are denoted as $T(e)$ and $H(e)$, respectively. We restrict the analysis to hypergraphs where all the hyperedges have only one source node $|T(e)| = 1$, and there is only one hyperedge per source node. This hypergraph model is sufficiently general to consider any accumulative wireless communication network with one node at each path position. There are many different notions of hyperpaths, see [14]. Here, we define a hyperpath as a sequence of nodes $\mathbf{p} = \langle \mathbf{p}[0], ..., \mathbf{p}[L+1] \rangle$ consisting of vertices $\mathbf{p}[i] \in V$. Let e_{u} denote the hyperedge associated to node u. A hyperpath exists if for every node in the path $\mathbf{p}[i]$, $0 < i \leq L+1$ there exits at least one preceding node $\mathbf{p}[j]$, $0 \leq j < i$, such that $\mathbf{p}[i] \in H\left(e_{\mathbf{p}[j]}\right)$.

To illustrate the existence condition of a hyperpath consider the hypergraph depicted in Fig. 4. Suppose there is no edge link connecting nodes A and B, and thus there can not be communication from A to B. Then, the path ⟨S,A,B,D⟩ does not exist. However, the hyperpath ⟨S,A,B,D⟩ exists, as there is a hyperedge link connecting nodes S and B, as well as, A and D. We may associate one metric to each of the vertices of a hyperedge, namely, $\beta_{\mathrm{u}} = \beta\left(e_{\mathrm{u}}\right) = \{\beta_{\mathrm{u,v}}, \forall \mathrm{v} \in H(e_{\mathrm{u}})\}$. Then, the weight of a hyperpath $w(\mathbf{p})$ is a function of the metrics of the edges traversed by the path, namely $w(\mathbf{p}) = w(\beta_{\mathbf{p}})$, where $\beta_{\mathbf{p}} = \left\{\beta_{\mathbf{p[i]}}, i = \{0, ..., L+1\}\right\}$. The hypergraph represented in Fig. 4 models the routing problem in AM wireless communications with two possible relays A and B. Observe that given that the source transmission is used by relay nodes A, B and the destination D, the hypereadge e_S connects node S to nodes A, B and D. Similarly, the hyperedges at relay nodes, e_A and e_B connect each of them to the other relay and destination.

III. Minimum Energy Accumulative Routing

The implications of accumulative communications in multi-hop routing problems are better understood by looking at specify examples of path weight functions. Here, we focus only on the minimum energy routing path weight function for a very simplistic linear energy allocation model. This example will be rich enought to disccuss in detail the need of hypergraph models. The path weight functions derived here will also be instrumental in subsequent sections.

We consider a simplistic accumulative communication model for wireless channels. The link between nodes u and v is modeled by the channel gain $g_{\mathrm{u,v}} \in \{\mathbb{R}^+, 0\}$. Let P_{u} denote the transmission power at node u, then the received signal power at node v is $g_{\mathrm{u,v}} P_{\mathrm{u}}$. A packet is correctly decoded at the destination node if the received signal power at the destination node exceeds a certain threshold level H_{D}. This model is valid for networks that operate in the wideband power limited regime. This regime is realistic for some wireless networks, such as sensor networks where there exist strong energy limitations at nodes, the traffic load is low, and there is sufficient large frequency bandwidth. Moreover, the analysis conducted here can be extended to any other scenario where the resource allocation problem is linear. Such linear dependence is forced here by considering energy accumulation in wideband signals, as in [3], or in [4], but it can also be found in other situations, such as when considering full-duplex relay terminals as in [8], or when optimizing over transmission durations instead of transmitted power as in [7] and [15].

Specifically, our network metric is the energy efficiency of a path measured as the ratio between the total transmitted power by all nodes in the path and the power received at destination to decode the message H_{D}. Accordingly, we define the minimum energy weight of the path $\mathbf{p}$ as

$$w(\mathbf{p}) = \frac{\sum_{i=0}^{L} P_{\mathbf{p}[i]}}{H_{\mathrm{D}}}.$$

In the following, we derive the minimum energy path weight function for different forwarding strategies at relays. We look for recursive path weight functions and distinguish between computing the weight of a path *forward* (from the source to the destination) or *backward* (from the destination to the source). More precisely, let $\mathbf{p}$ represent a *path* in the direction of the communication, this is, from the source node S $= \mathbf{p}[0]$ to the destination node D $= \mathbf{p}[L+1]$. We define the *forward partial path* from the source node S= $\mathbf{p}[0]$ to relay node $\mathbf{p}[i]$, as $\mathbf{p}_i = \langle \mathbf{p}[0], \mathbf{p}[j+1], ..., \mathbf{p}[i] \rangle$, and the *backward partial path* from node $\mathbf{p}[L+1-i]$ to the destination node D=$\mathbf{p}[L+1]$ as $\overleftarrow{\mathbf{p}}_i$= $\langle \mathbf{p}[L+1-i], \mathbf{p}[L+1-(i-1)], ..., \mathbf{p}[L+1] \rangle$. We say that the path weight function $w(\mathbf{p})$ is computed *forward* if we compute the weights of the forward partial paths $w(\mathbf{p}_i)$, recursively, for $i = 0, ..., L+1$. Similarly, we say that the path weight is computed *backward* when we compute the

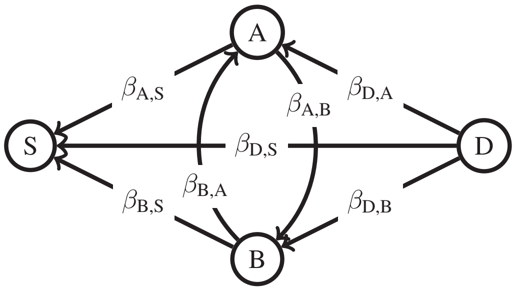

Fig. 5. DF TM and PF AM (backward) graph model ($N = 4$). For TM $\beta_{\text{u,v}} = h_{\text{u,v}} = g_{\text{v,u}}$. For PF AM $\beta\left(e_{\text{u,v}}\right) = \{h_{\text{u,v}}, h_{\text{D,v}}\} = \{g_{\text{v,u}}, g_{\text{v,D}}\}$.

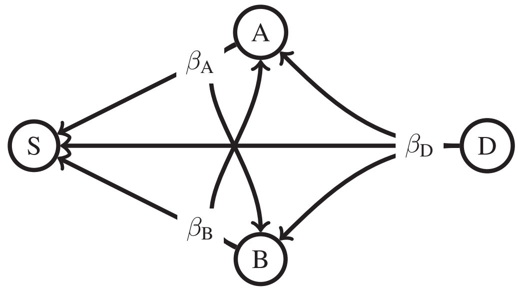

Fig. 6. AM (backward) hypergraph model ($N = 4$). $\beta_{\text{D}} = \{h_{\text{D,S}}, h_{\text{D,B}}, h_{\text{D,A}}\}$, $\beta_{\text{A}} = \{h_{\text{A,B}}, h_{\text{A,S}}\}$, $\beta_{\text{B}} = \{h_{\text{B,A}}, h_{\text{B,S}}\}$.

weights of the backward partial paths $w(\overleftarrow{\mathbf{p}}_i)$, recursively, for $i = 0, ..., L+1$. Depending on the forwarding strategy adopted, we will find a forward, a backward or both recursive path weight functions. Iterative path search algorithms such as Dijkstra's algorithm find paths from a source to any destination using the forward path weight function, and from any source to a given destination using the backward path weight function.

For the sake of notation simplicity, when there is no ambiguity on which forward path $\mathbf{p}$ we are referring to, we denote $g_{i,j} = g_{\mathbf{p}[i],\mathbf{p}[j]}$, and $P_i = P_{\mathbf{p}[i]}$. Similarly, for a backward path $\overleftarrow{\mathbf{p}}$, we denote $h_{i,j} = h_{\overleftarrow{\mathbf{p}}[i],\overleftarrow{\mathbf{p}}[j]}$ and $P_i = P_{\overleftarrow{\mathbf{p}}[i]}$. Given that $\overleftarrow{\mathbf{p}}[i] = \mathbf{p}[L+1-i]$, we have $h_{j,i} = g_{L+1-i,L+1-j}$.

A. DF TM network

In a TM communication network, all nodes in the path $\mathbf{p}$ decode the source message, and thus, not only the destination but also all relays must receive a power exceeding H_D from the immediately previous relay node. Consequently, given a path $\mathbf{p}$, the minimum power transmitted by every node $\mathbf{p}[i]$ satisfies $H_{\text{D}} = g_{i,i+1}P_i$. Accordingly, the weight of the partial path $\mathbf{p}_1 = \langle \mathbf{p}[0], \mathbf{p}[1] \rangle$ between the source and the first relay is given by $w(\mathbf{p}_1) = \frac{P_0}{H_{\text{D}}} = \frac{1}{g_{0,1}}$. In the next hop, the power transmitted by node $\mathbf{p}[1]$ must satisfy $H_{\text{D}} = g_{1,2}P_1$, and thus, the weight of the partial path $\mathbf{p}_2$ is given by

$$w(\mathbf{p}_2) = \frac{P_0 + P_1}{H_{\text{D}}} = \frac{1}{g_{1,2}} + w(\mathbf{p}_1).$$

Repeating this argument for every hop, it can be shown that the weight of the partial path $\mathbf{p}_i$ can be computed as

$$w(\mathbf{p}_i) = \frac{1}{g_{i-1,i}} + w(\mathbf{p}_{i-1}). \tag{2}$$

Similarly, we could have obtained a backward counter part as

$$w(\overleftarrow{\mathbf{p}}_i) = \frac{1}{h_{i-1,i}} + w(\overleftarrow{\mathbf{p}}_{i-1}) \tag{3}$$

where $h_{j,i} = g_{L+1-i,L+1-j}$. Observe that both path weigh function recursion are equivalent $w\left(\overleftarrow{\mathbf{p}}_{L+1}\right) = w\left(\mathbf{p}_{L+1}\right)$.

To find paths in a TM wireless network from a given source to a destination, we use the forward path weight function (2), together with the fully connected graph illustrated in Fig. 2, where the metric associated to the edge $e_{\text{u,v}}$ is directly given by the channel gain $\beta(e_{\text{u,v}}) = g_{\text{u,v}}$. Instead, to find paths in a TM wireless network to a given destination node from a source, we can use the backward path weight function, together with the fully connected graph illustrated in Fig. 5 where the metric associated to the edge $e_{\text{u,v}}$ is given by the backward channel gain $\beta(e_{\text{u,v}}) = h_{\text{u,v}} = g_{\text{v,u}}$.

B. DF AM network

The TM network can be seen as a particular case of the DF AM network without accumulative capabilities at nodes. In the most general DF AM network, relay nodes obtain multiple energy leakages from previous transmissions, and thus accumulate energy/information from them. As for the DF TM network, relay nodes must decode the source message before transmission, and thus, they need to receive an aggregated signal power exceeding H_{D}[1]. Accordingly, the minimum power transmitted by the source, P_0, so that the first relay $\mathbf{p}[1]$ decodes the source message, must satisfy

$$H_{\text{D}} = g_{0,1}P_0 \tag{4}$$

and thus $w(\mathbf{p}_1) = \frac{1}{g_{0,1}}$. However, given that the second relay $\mathbf{p}[2]$ has already received the power $g_{0,2}P_0$ from the source, the minimum power that node $\mathbf{p}[1]$ must transmit so that node $\mathbf{p}[2]$ decodes the source message must satisfy

$$H_{\text{D}} = g_{1,2}P_1 + g_{0,2}P_0. \tag{5}$$

Equating (4) and (5), we have

$$P_1 = \frac{g_{0,1} - g_{0,2}}{g_{1,2}} P_0. \tag{6}$$

Thus, the weight of the partial path $w(\mathbf{p}_2)$ is given by

$$\begin{aligned} w(\mathbf{p}_2) &= \frac{P_0 + P_1}{H_{\text{D}}} = \frac{1}{g_{0,1}} + \frac{1}{g_{1,2}}\left(1 - \frac{g_{0,2}}{g_{0,1}}\right) \\ &= \frac{1}{g_{1,2}} + \frac{g_{1,2} - g_{0,2}}{g_{1,2}} w(\mathbf{p}_1). \end{aligned}$$

Observe from (6) that if $g_{0,2} > g_{0,1}$, or, equivalently, if $w(\mathbf{p}_2) < w(\mathbf{p}_1)$, then the relay node $\mathbf{p}[2]$ has already received enough power from the source node, and thus enforcing (5) implies $P_1 < 0$, which is not a feasible solution. To avoid non

[1]For the DF AM, we assume that the decoding order coincides with the transmission path, this is, among all nodes in a given path, the next node in transmitting must be also the next node in decoding the message. For the minimum energy routing problem considered here, this can be assume without loss of optimality as shown in [4, Theorem 3]. For other routing metrics, and in more general situations such as if simultaneous transmissions from multiple nodes are beneficial, the decoding order and the transmission path problems become coupled and must be jointly solved, see [16] and references therein for recent advances in more general settings.

feasible paths, we set $w(\mathbf{p}_2)=\infty$ if $w(\mathbf{p}_2)<w(\mathbf{p}_1)$. Then, $w(\mathbf{p}_i)$ can be computed recursively as

$$w(\mathbf{p}_i)=\frac{1}{g_{i-1,i}}+\sum_{j=1}^{i-1}\frac{g_{j,i}-g_{j-1,i}}{g_{i-1,i}}w(\mathbf{p}_j) \tag{7}$$

if $w(\mathbf{p}_i)\geq w(\mathbf{p}_{i-1})$, and $w(\mathbf{p}_i)=\infty$ otherwise.

Defining $h_{i,j}=g_{L+1-j,L+1-i}$, we derive in the Appendix the backward path weight function

$$w(\overleftarrow{\mathbf{p}}_i)=\frac{1}{h_{i-1,i}}+\sum_{j=1}^{i-1}\frac{h_{j,i}-h_{j-1,i}}{h_{i-1,i}}w(\overleftarrow{\mathbf{p}}_j) \tag{8}$$

if $w(\overleftarrow{\mathbf{p}}_i)\geq w(\overleftarrow{\mathbf{p}}_{i-1})$, and $w(\overleftarrow{\mathbf{p}}_i)=\infty$ otherwise. In contrast to the forward path computation, in the backward computation satisfying $w(\overleftarrow{\mathbf{p}}_i)\geq w(\overleftarrow{\mathbf{p}}_{i-1})$ does not imply $P_i\geq 0$, and thus, we can not guarantee the feasibility of a path at each iteration. This makes challenging the backward path search in DF AM networks. However, we have shown in [13] that the lightest path $\overleftarrow{\mathbf{p}}_i^\star$ according to the backward path weight function is always forward feasible, this is $w(\mathbf{p}_i^\star)\geq w(\mathbf{p}_{i-1}^\star)$ or, equivalently $P_i\geq 0$, where $\mathbf{p}^\star[i]=\overleftarrow{\mathbf{p}}^\star[L+1-i]$ and thus, as long as, we use an optimal algorithm that finds the lightest path, we can use the backward path weight function in (8).

Observe that the forward path weight functions can not be computed over directed graphs, as the one depicted in Fig. 2, by associating the channel gain metric $\beta(e_{\text{u,v}})=g_{\text{u,v}}$ to the edge $e_{\text{u,v}}$ as we have done previously for the TM network. Consider, for instance, the computation of the weight of the path ⟨S,A,D⟩. The weight of the path ⟨S,A⟩ is given by $w(\langle\text{S,A}\rangle)=\frac{1}{g_{\text{S,A}}}$. Then, the weight of the path ⟨S,A,D⟩, $w(\langle\text{S,A,D}\rangle)=\frac{1}{g_{\text{A,D}}}+\frac{g_{\text{A,D}}-g_{\text{S,D}}}{g_{\text{A,D}}}w(\langle\text{S,A}\rangle)$, depends on $g_{S,D}$ which is a metric not associated to the path ⟨A,D⟩. For the AM network, we need to described the network using a hypergraph, as the one shown in Fig. 4, where the metric associated to the edge of node u includes the channel gains from node u to all the network nodes, namely $\beta(e_{\text{u}})=\{g_{\text{u,v}},\forall\text{v}\}$. Similarly, we need to use the hypergraph in Fig. 6 for the backward path weight function.

C. Cut-set bound

In the AM communication network, asking every node to decode the source message is not always needed. Relays can, for example, amplify or compress and forward the received signals, without decoding the information. We can have an idea of the path weight functions that may appear with these non-regenerative relaying strategies by looking at the cut-set bound. Indeed in [17] it is shown that the cut-set bound rates can be achieved within a constant rate gap by compress and forward like strategies. For the CB, we remove the decoding constraint at relay nodes. We assume that only the destination decodes the message by receiving an aggregated power exceeding H_{D}. At relays, the input-output power flow is such that the total received power at a node equals the total output received power from that node. This is, the power transmitted by node $\mathbf{p}[i]$ is such that subsequent nodes in the path $\mathbf{p}[j]$, $j>i$ receive an aggregated power equal to the power that node $\mathbf{p}[i]$ has received from previous nodes $\mathbf{p}[j]$, $j<i$. This is,

$$P_i\sum_{j=i+1}^{L+1}g_{i,j}=\sum_{j=0}^{i-1}P_j g_{j,i}. \tag{9}$$

or, equivalently, using the backward path ordering

$$P_i\sum_{i=0}^{i-1}h_{j,i}=\sum_{j=i-1}^{L+1}P_j h_{i,j}.$$

In this case, it is more convenient to obtain the backward path weight function. Recall that the destination node is referred to as D$=\overleftarrow{\mathbf{p}}[0]$, the source as S$=\overleftarrow{\mathbf{p}}[L+1]$, and the path from node i to the destination is denoted as $\overleftarrow{\mathbf{p}}_i=\langle\mathbf{p}[i],...,\mathbf{p}[0]\rangle$.

Consider first the computation of the weight of the path $\overleftarrow{\mathbf{p}}_1=\langle\overleftarrow{\mathbf{p}}[1],\overleftarrow{\mathbf{p}}[0]\rangle$. Suppose node $\overleftarrow{\mathbf{p}}[1]$ is the source of the communication. In this case, node $\overleftarrow{\mathbf{p}}[1]$ transmits with power P_1 and destination receives $H_D=h_{0,1}P_1$, then

$$w(\overleftarrow{\mathbf{p}}_1)=\frac{P_1}{H_D}=\frac{1}{h_{0,1}}.$$

Consider next the computation of the weight of the path $\overleftarrow{\mathbf{p}}_2=\langle\overleftarrow{\mathbf{p}}[2],\overleftarrow{\mathbf{p}}[1],\overleftarrow{\mathbf{p}}[0]\rangle$. Suppose node $\overleftarrow{\mathbf{p}}[2]$ is now the source of the communication. The received aggregated power at destination is $H_D=h_{0,1}P_1+h_{0,2}P_2$. Thus, the weight of the partial path $\overleftarrow{\mathbf{p}}_2$ is

$$w\left(\overleftarrow{\mathbf{p}}_2\right)=\frac{P_1+P_2}{h_{0,1}P_1+h_{0,2}P_2}. \tag{10}$$

By enforcing the input-output power flow condition in (9) with $L=1$, we require that the power received at node $\overleftarrow{\mathbf{p}}[1]$ from node $\overleftarrow{\mathbf{p}}[2]$, equals the power received at destination from node $\overleftarrow{\mathbf{p}}[1]$, namely $h_{1,2}P_2=P_1h_{0,1}$. Substituting the later equality into (10), we obtain

$$w\left(\overleftarrow{\mathbf{p}}_2\right)=\frac{1+h_{1,2}w\left(\overleftarrow{\mathbf{p}}_1\right)}{h_{1,2}+h_{0.2}}.$$

Similarly, for the partial path $\overleftarrow{\mathbf{p}}_3$, the destination receives an aggregated power of $h_{0,1}P_1+h_{0,2}P_2+h_{0,3}P_3$, and the input-output power flow for $L=3$ requires

$$\begin{aligned}P_1h_{0,1}&=h_{1,2}P_2+h_{1,3}P_3,\\ P_2\left(h_{0,2}+h_{1,2}\right)&=h_{2,3}P_3.\end{aligned}$$

The weight of the partial path $\mathbf{p}_{0,3}$ is then given by

$$w\left(\overleftarrow{\mathbf{p}}_3\right)=\frac{1+h_{1,3}w\left(\overleftarrow{\mathbf{p}}_1\right)+h_{2,3}w\left(\overleftarrow{\mathbf{p}}_2\right)}{h_{2,3}+h_{1,3}+h_{0,3}}.$$

In general, it can be shown that the weight of the partial path $\overleftarrow{\mathbf{p}}_i$ can be computed recursively as

$$w\left(\overleftarrow{\mathbf{p}}_i\right)=\frac{1+\sum_{j=1}^{i-1}h_{j,i}w\left(\overleftarrow{\mathbf{p}}_j\right)}{\sum_{j=0}^{i-1}h_{j,i}}. \tag{11}$$

Observe that we can compute this path weight over the hypergraph model in Fig. 6 where the metric associated to the hyperedges of each node is given by: $\beta_{\text{D}}=\{h_{\text{D,S}},h_{\text{D,B}},h_{\text{D,A}}\}$, $\beta_{\text{A}}=\{h_{\text{A,B}},h_{\text{A,S}}\}$, $\beta_{\text{B}}=\{h_{\text{B,A}},h_{\text{B,S}}\}$.

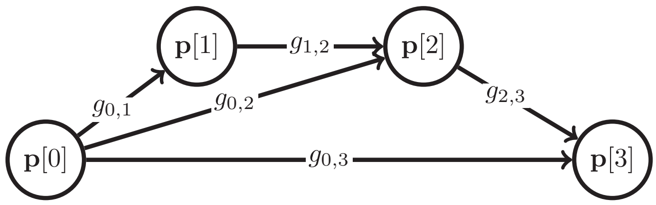

Fig. 7. SAM communication model.

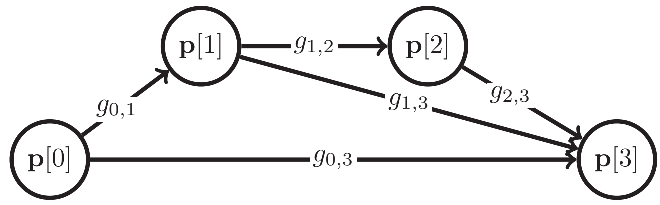

Fig. 9. DAM communication model.

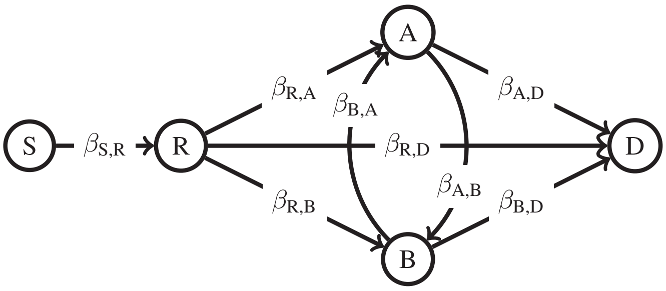

Fig. 8. DF SAM graph model ($N = 5$, $\beta_{\text{S,R}} = g_{\text{S,R}}$,$\beta_{\text{u,v}} = \frac{g_{\text{u,v}} g_{\text{S,R}}}{g_{\text{S,R}} - g_{\text{S,v}}}$).

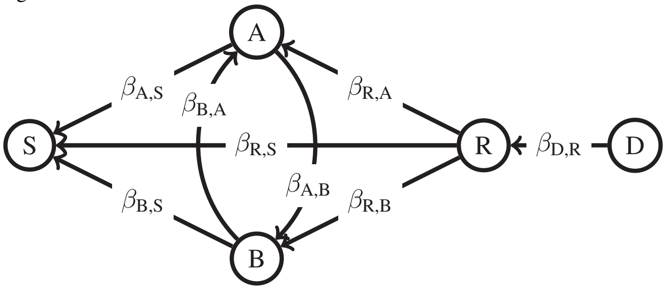

Fig. 10. DF DAM graph model ($N = 5$, $\beta_{\text{D,R}} = h_{\text{D,R}}$,$\beta_{\text{u,v}} = \frac{h_{\text{u,v}} h_{\text{D,R}}}{h_{\text{D,R}} - h_{\text{D,v}}}$).

D. AM networks over graphs

Observe that we could have obtained the DF TM path weight functions in (2) and (3) by particularizing the DF AM path weight functions in (7) and (8), respectively, for the case where only the channel gains between consecutive nodes are not null, namely $g_{j,i} = 0, \forall j < i - 1$. By removing all the accumulative capabilities, we have shown that, the TM path weight can be computed over a graph. Motivated by the TM network situation, we here look for other non trivial situations where AM networks can be represented over graphs while keeping energy accumulation at some nodes. If we can model the routing problem using a graph then we can use known optimality conditions [2] to find optimal path search algorithms.

1) DF SAM network: We consider first the Source Accumulative Multi-hop (SAM) communication network depicted in Fig. 7. In the SAM network, relay node $\mathbf{p}[i]$ only accumulates the received power from the source node $\mathbf{p}[0]$ and the immediately previous node $\mathbf{p}[i-1]$. The power received from the rest of nodes $\mathbf{p}[j]$ $0 < j < i-1$ is ignored. This situation, is equivalent to setting $g_{j,i} = 0$ for $0 < j < i - 1$ in the DF AM network. Thus, particularizing the forward path weight function in (7), we obtain

$$\begin{aligned} w(\mathbf{p}_i) &= \frac{1}{g_{i-1,i}} + \frac{g_{i-1,i} w(\mathbf{p}_{i-1}) - g_{0,i} w(\mathbf{p}_1)}{g_{i-1,i}} \\ &= \frac{g_{0,1} - g_{0,i}}{g_{i-1,i} g_{0,1}} + w(\mathbf{p}_{i-1}) \end{aligned} \tag{12}$$

if $w(\mathbf{p}_i) > w(\mathbf{p}_{i-1})$ and $w(\mathbf{p}_i) = \infty$, otherwise. This path weight function can now be computed using the directed graph illustrated in Fig. 8 where not only the source S and destination D but also the first relay R are fixed, and we define the metric associated to the edge between nodes u, and v, as $\beta\left(e_{\text{u,v}}\right) = \beta_{\text{u,v}} = \frac{g_{\text{u,v}} g_{\text{S,R}}}{g_{\text{S,R}} - g_{\text{S,v}}}$. By using this path weight metric, we can search for the best path from any source-relay pair $\langle$S,R$\rangle$ to every other network node using a graph instead of a hypergraph. We however need to repeat the search for all possible first relay nodes, R, to obtain the lightest path from the source node S.

2) DF DAM Network: Next, we consider the somewhat dual DF Destination Acumulative Multi-hop (DAM) comunication network depicted in Fig. 9. In the DAM network relays only listen to the immediately previous node, whereas the destination accumulates the signals received from all previous transmissions. For the example in Fig. 3, this situation can be modeled by assuming ($g_{0,2} = 0$) and in general assuming $g_{j,i} = 0$ if $j \neq i - 1$ or $i \neq L + 1$, which is equivalent to $h_{j,i} = 0$ for $0 < j < i - 1$. Particularizing the DF backward path weight function in (8), we can obtain the backward path weight function as

$$\begin{aligned} w(\overleftarrow{\mathbf{p}}_i) &= \frac{1}{h_{i-1,i}} + \frac{h_{i-1,i} w(\overleftarrow{p}_{i-1}) - h_{0,i} w(\overleftarrow{p}_1)}{h_{i-1,i}} \\ &= \frac{h_{0,1} - h_{0,i}}{h_{i-1,i} h_{0,1}} + w(\mathbf{p}_{i-1}) \end{aligned} \tag{13}$$

if $w(\mathbf{p}_i) > w(\mathbf{p}_{i-1})$ and $w(\mathbf{p}_i) = \infty$, otherwise. This path weight function can now be computed using the directed graph illustrated in Fig. 10 to find the lightest path in a backward manner. In Fig. 10, not only the source S and destination D but also the last relay R are fixed, and we define the metric associated to the edge between nodes u, and v, as $\beta\left(e_{\text{u,v}}\right) = \beta_{\text{u,v}} = \frac{h_{\text{u,v}} h_{\text{D,R}}}{h_{\text{D,R}} - h_{\text{D,v}}}$. As discussed for the backward path weigh function of the DF AM network, as long as, we guarantee the optimality of the path search algorithm, we can use the backward path weight function since the lightest path $\mathbf{p}^\star$ is always feasible, this is $P_i > 0$ for all $\mathbf{p}^\star[i]$.

3) PF DAM network: For the DAM communication network, we also consider the parity forwarding (PF) scheme introduced in [18]. In contrast to the DF forwarding strategy, for the PF scheme, relay nodes do not need to recover the complete source message. This alleviates the decoding constraints at relays and leads to higher rates for DAM networks.

The basic principle of PF is the following. Consider the backward path $\overleftarrow{\mathbf{p}}$. Recall that, in the direction of the communication, node $\overleftarrow{\mathbf{p}}[i+1]$ transmits previous to node $\overleftarrow{\mathbf{p}}[i]$. The source node S $= \overleftarrow{\mathbf{p}}[L+1]$ transmits the message s_{L+1}, relays instead transmits only parity bits, s_i, of the parity bits message transmitted by the immediately previous node. The destination successively recovers all parity information to finally obtain the source message. Suppose node $\overleftarrow{\mathbf{p}}[i+1]$ broadcasts a message s_{i+1} which can be decoded by the next relay node $\overleftarrow{\mathbf{p}}[i]$ as long as it receives a power exceeding H_{i+1}. After decoding this message node $\overleftarrow{\mathbf{p}}[i]$ computes the parity bits, s_i, and broadcasts them to the next relay node and to the destination. Suppose message s_i can be decoded by receiving the signal from node $\overleftarrow{\mathbf{p}}[i]$ with a power exceeding H_i with $H_i \leq H_{i+1}$. Then, it follows from [18] that if a node (in particular the destination node) decodes the parity bits sent by node $\overleftarrow{\mathbf{p}}[i]$, then it can also decode the message transmitted by node $\overleftarrow{\mathbf{p}}[i+1]$ as long as the power received from node $\overleftarrow{\mathbf{p}}[i+1]$ exceeds H_{i+1}-H_i instead of H_{i+1}.

More specifically, let us first consider the transmission of the parity information, s_1, from the last relay $\overleftarrow{\mathbf{p}}[1]$ to the destination $\overleftarrow{\mathbf{p}}[0]$. Suppose this message requires a received power exceeding H_1 to be decoded. Then, the transmitted signal power from the last relay P_1 must satisfy $H_1 = h_{0,1}P_1$, and thus the weight of the partial path $\overleftarrow{\mathbf{p}}_1 = \langle \overleftarrow{\mathbf{p}}[1], \overleftarrow{\mathbf{p}}[0] \rangle$ is

$$w(\overleftarrow{\mathbf{p}}_1) = \frac{P_1}{H_1} = \frac{P_1}{h_{0,1}P_1} = \frac{1}{h_{0,1}}.$$

Recall that the message s_1 contains the parity bits of the message s_2 transmitted from node $\overleftarrow{\mathbf{p}}[2]$. Suppose that message s_2 requires a receive power exceeding H_2 to be decoded. Node $\overleftarrow{\mathbf{p}}[1]$ must have decoded the message s_2 from $\overleftarrow{\mathbf{p}}[2]$ in order to compute the parity information s_1, and thus, we require $H_2 = h_{1,2}P_2$. The destination node first decodes the parity information, s_1, transmitted from relay node $\overleftarrow{\mathbf{p}}[1]$ if $H_1 = h_{0,1}P_1$. Then, by using s_1, the destination obtains s_2 if the received signal from $\overleftarrow{\mathbf{p}}[2]$ satisfies $H_2 - H_1 = h_{0,2}P_2$. Combining these three equalities, we have $P_2 = \frac{H_1}{h_{1,2}-h_{0,2}}$, and thus, $P_2 \geq 0$ if and only if $h_{0,2} \leq h_{1,2}$. Then, the weight of the partial path $\overleftarrow{\mathbf{p}}_2 = \langle \overleftarrow{\mathbf{p}}[2], \overleftarrow{\mathbf{p}}[1], \overleftarrow{\mathbf{p}}[0] \rangle$ is given by

$$w\left(\overleftarrow{\mathbf{p}}_2\right) = \frac{P_1+P_2}{H_2} = \frac{1}{h_{1,2}} + \left(1 - \frac{h_{0,2}}{h_{1,2}}\right) w\left(\overleftarrow{\mathbf{p}}_1\right) \quad (14)$$

if $h_{1,2} \geq h_{0,2}$ and $w\left(\overleftarrow{\mathbf{p}}_2\right) = \infty$ otherwise.

Following the same argument, in order to obtain the parity bits s_2, node $\overleftarrow{\mathbf{p}}[2]$ must receive sufficient power from node $\overleftarrow{\mathbf{p}}[3]$ to decode s_3, and thus we require $H_3 = h_{2,3}P_3$. Then, from (14), we have that the destination can recover the parity message sent by node $\overleftarrow{\mathbf{p}}[2]$ if $H_2 w\left(\overleftarrow{\mathbf{p}}_2\right) = P_1 + P_2$. Using the parity information s_2, the destination can finally decode the message s_3 if the power received from node $\overleftarrow{\mathbf{p}}[3]$ satisfies $H_3 - H_2 = h_{0,3}P_3$. Combining these three equalities, we can show that $P_3 = \frac{H_2}{h_{2,3}-h_{0,3}}$, and, thus $P_3 \geq 0$ if and only if $h_{0,3} \leq h_{2,3}$. Then, the weight of the partial path $\overleftarrow{\mathbf{p}}_3 = \langle \overleftarrow{\mathbf{p}}[3], \overleftarrow{\mathbf{p}}[2], \overleftarrow{\mathbf{p}}[1], \overleftarrow{\mathbf{p}}[0] \rangle$ is given by

$$w\left(\overleftarrow{\mathbf{p}}_3\right) = \frac{P_1+P_2+P_3}{H_3} = \frac{1}{h_{2,3}} + \left(1 - \frac{h_{0,3}}{h_{2,3}}\right) w\left(\overleftarrow{\mathbf{p}}_2\right)$$

if $h_{1,3} \geq h_{0,3}$ and $w\left(\overleftarrow{\mathbf{p}}_3\right) = \infty$ otherwise. Finally, it can be shown that the weight of the partial path $\overleftarrow{\mathbf{p}}_i$ is given by

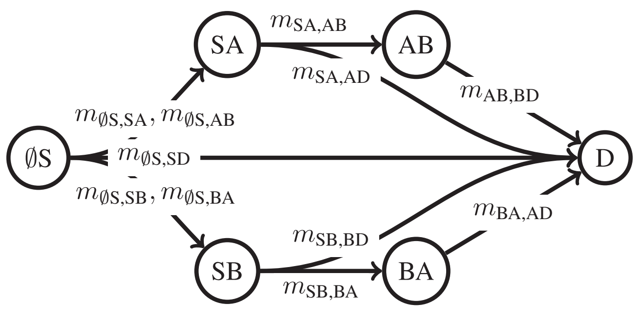

Fig. 11. DF Extended forward AM (EAM) hypergraph model ($N = 4, m_{\text{uv,wx}} = g_{\text{u,v}} - g_{\text{u,x}}$).

$$w\left(\overleftarrow{\mathbf{p}}_i\right) = \frac{1}{h_{i-1,i}} + \left(1 - \frac{h_{0,i}}{h_{i-1,i}}\right) w\left(\overleftarrow{\mathbf{p}}_{i-1}\right) \quad (15)$$

if $h_{i-1,i} \geq h_{0,i}$ and $w\left(\overleftarrow{\mathbf{p}}_i\right) = \infty$ otherwise. Observe, that for the PF strategy the feasibility of the path $P_i \geq 0$ can be guaranteed at each iteration.

To find paths from a source to a given destination in a PF AM network, we can use the backward path weight (15) together with the fully connected graph illustrated in Fig. 5 for $N = 4$, where the metric associated to the edge $e_{\text{u,v}}$ is given by $\beta(e_{\text{u,v}}) = \{h_{\text{u,v}}, h_{\text{D,v}}\} = \{g_{\text{v,u}}, g_{\text{v,D}}\}$.

E. DF Extended AM Network

It is well known that the minimum energy routing problem for the DF AM network is NP complete and thus, there is no optimal path search algorithm for this network. Using the optimality conditions derived in Section IV for the Dijkstra's path search algorithm in hypergraphs, we will be able to show in Section V that Dijkstra's algorithm is optimal for the CB path weight function in (11). Motivated by this success, we here transforms the AM hypergraph into an Extended AM (EAM) hypergraph for which the DF AM path weight function reads as the CB path weight function. In the extended hypergraph, the path weight function will satisfy the optimality conditions for Dijkstra's algorithm in certain situations.

We illustrate the extended hypergraph model in Fig. 11 for a network with 2 available relays $N = 4$. In addition to the source node S and destination node D, each possible combination of a pair of relay nodes u and v, u≠v (excluding destination node) in the original hypergraph is represented by a single node uv in the extended hypergraph. The hyperedge associated to node uv, e_{uv} connects node uv to nodes $\{[\text{wx}], \text{v=w}, \text{x} \neq \text{u}\}$. The metric associated to this hypereadge is given by $\beta(e_{\text{uv}}) = \{m_{\text{uv,wx}}, \text{x} \neq \text{u}\}$. Observe that $m_{\text{uv,wx}} = m_{\text{uv,yx}}$ for all $\text{w} \neq \text{y}$. In the case of a fully connected communication model, we transform the original fully connected hypergraph with N nodes, to a partially connected hypergraph with $(N-2)^2 + 2$ nodes.

Given a path in the extended hypergraph $\mathbf{e} = \langle \mathbf{e}[0], \mathbf{e}[1], ..., \mathbf{e}[L+1] \rangle$. For each node $\mathbf{e}[i]$ in the extended hypergraph, we identify the two nodes of the original hypergraph as $\mathbf{e}[i]\{1\}$ and $\mathbf{e}[i]\{2\}$. Then, we can recover the transmission path simply as $\mathbf{p}[i] = \mathbf{e}[i]\{2\}$, $i = 0, ..., L+1$, or as $\mathbf{p}[i] = \mathbf{e}[i+1]\{1\}$, $i = 0, ..., L$. Observe that if we define

$m_{\mathbf{e}[j],\mathbf{e}[i]} \triangleq g_{\mathbf{e}[j]\{2\},\mathbf{e}[i]\{2\}} - g_{\mathbf{e}[j]\{1\},\mathbf{e}[i]\{2\}}$ for $0 < j < i$, and $m_{\mathbf{e}[0],\mathbf{e}[i]} \triangleq g_{\mathbf{e}[j]\{2\},\mathbf{e}[i]\{2\}}$, then we can write

$$\begin{aligned} m_{j,i} &= m_{\mathbf{e}[j],\mathbf{e}[i]} && \text{(16a)} \\ &= g_{\mathbf{e}[j]\{2\},\mathbf{e}[i]\{2\}} - g_{\mathbf{e}[j]\{1\},\mathbf{e}[i]\{2\}} && \text{(16b)} \\ &= g_{\mathbf{p}[j],\mathbf{p}[i],} - g_{\mathbf{p}[j-1],\mathbf{p}[i]} && \text{(16c)} \\ &= g_{j,i} - g_{j-1,i} && \text{(16d)} \end{aligned}$$

and $m_{0,i} = g_{0,i}$. Then, since $g_{i-1,i} = \sum_{j=0}^{i-1} m_{j,i}$, the path weight function in (7) reads as the CB path weight

$$w(\mathbf{p}_i) = \frac{1 + \sum_{j=1}^{i-1} m_{j,i} w(\mathbf{p}_j)}{\sum_{j=0}^{i-1} m_{j,i}}. \quad (17)$$

We will see in Section V that to satisfy the Dijkstra's optimality conditions we need $m_{j,i} \geq 0$. Consequently, we can only guarantee finding the best path among those satisfying satisfying $m_{j,i} \geq 0$ or, equivalently, $g_{j,i} \geq g_{j-1,i}$ for all $l < j < i$. This is, we only look for paths $\mathbf{p}$ satisfying that the channel gain to node $\mathbf{p}[i]$ from all previous nodes in the path $\mathbf{p}[l]$, $g_{l,i}$ increases monotonically for $l = 0, ..., i-1$. As we will show in simulation results the optimal path is found inside this subset of paths with high probability. Moreover, notice that in contrast to the DF SAM, DF DAM and PF DAM networks, we can keep all the accumulative capabilities at nodes, and thus even if we use a suboptimal path, the path weight achieved (energy efficiency) is very close to that of the optimal path.

IV. Optimality of Dijkstra's Algorithm in Accumulative Networks

In previous section, we have provided the different path weight functions that we have encountered when addressing the problem of finding minimum energy paths in static accumulative multi-hop networks. We have seen that in some cases the path weight functions can be computed over graphs as for the TM, DF SAM, DF DAM and PF DAM networks, however in general, we have required hypergraphs to compute path weights function in AM networks. Here, we address the problem of finding the conditions that guarantee that a certain path search algorithm finds the lightest path in accumulative multi-hop networks. Whenever the routing problem can be represented by a graph, the conditions that guarantee the optimality of Bellman-Ford and Dijkstra's algorithms can be found in [2]. Here, we limit the discussion to the Dijkstra's path search algorithm over the directed hypergraphs $\mathcal{H}(V, E)$ introduced in previous section. To that end, we first review the well known optimality conditions for Dijkstra's algorithm in graphs and discuss their extensions to hypergraphs. We show that the resultant conditions are only sufficient but not necessary for optimality. This is shown by providing a new set of sufficient conditions for optimality. In the next section, we use these new conditions to prove the optimality of Dijkstra's algorithm for the CB and DF EAM path weight functions.

We begin by providing the mathematical representation of a path selection criteria which is usually called as routing metric. We represent a routing metric following the notation in [2] as an algebra on top of a quadruplet $(Q, \oplus, w, \preceq)$, where Q is the set of all possible paths, $\oplus$ is a binary operation that maps pairs with a path and an ordered sequence of nodes into a path, i.e. if the path $\mathbf{a} \in Q$ and the last node in $\mathbf{a}$ coincides with the first node of the ordered sequence of nodes $\mathbf{b}$, then $\mathbf{a} \oplus \mathbf{b}$ denotes the concatenation of path $\mathbf{a}$ with the ordered sequence of nodes $\mathbf{b}$, with $\mathbf{a} \oplus \mathbf{b} \in Q$, w is a function that maps a path to a weight, and $\preceq$ is an order relation, where $w(\mathbf{a}) \preceq w(\mathbf{b})$ means the path $\mathbf{a}$ is lighter (better) that or equal to $\mathbf{b}$. Given a routing metric $(Q, \oplus, w, \preceq)$, a routing protocol operates with the path weights of the paths in Q to find the lightest path $\mathbf{q}^* \in Q$ between a source and a destination.

The concatenation operation as defined above differs slightly from the one defined in [2] for graphs. In [2], $\oplus$ concatenates two paths in Q, and returns a path also in Q. The definition of $\oplus$ presented here is motivated by the fact that in a hypergraph, even if the ordered set of nodes $\mathbf{b}$ does not belong to Q, the path $\mathbf{a} \oplus \mathbf{b}$ might belong to Q.

A. Extension of Dijkstra's optimality conditions in graphs

Here we review the conditions that guarantee that Dijkstra's algorithm finds the lightest path in a directed graph $\mathcal{G}(V, E)$, and discuss their extension to directed hypergraphs $\mathcal{H}(V, E)$.

Given a graph, [1] and [2] developed a comprehensive framework to identify the specific conditions a routing metric needs to satisfy in order to be combined with a certain type of optimal routing protocol to obtain the optimal path. In particular, it was shown that Dijkstra's algorithm with source routing is optimal if and only if, the routing metric satisfies right-monotonicity and right-isotonicity. These properties are here stated, mostly, as they appear in [2] with the necessary modifications to account for the new definition of the binary operation $\oplus$.

Definition 1. *The quadruplet* $(Q, \oplus, w, \preceq)$ *is* right-monotonic *if* $w(\mathbf{a}) \preceq w(\mathbf{a} \oplus \mathbf{b})$, *for any paths* $\mathbf{a}$ *and* $\mathbf{a} \oplus \mathbf{b}$ *in* Q.

Definition 2. *Given the paths* $\mathbf{a}$ *and* $\mathbf{b}$ *between two nodes A and B, and the paths* $\mathbf{a} \oplus \mathbf{c}$ *and* $\mathbf{b} \oplus \mathbf{c}$ *from A to a third node C, sharing the nodes in* $\mathbf{c}$. *If* $w(\mathbf{a}) \preceq w(\mathbf{b})$, *the quadruplet* $(Q, \oplus, w, \preceq)$ *is right-isotonic if* $w(\mathbf{a} \oplus \mathbf{c}) \preceq w(\mathbf{b} \oplus \mathbf{c})$ *for any paths* $\mathbf{a}$, $\mathbf{b}$, $\mathbf{a} \oplus \mathbf{c}$, $\mathbf{b} \oplus \mathbf{c}$ *in* Q.

Our definition of right-monotonicity differs from the one in [2] in that it does not restrict the path $\mathbf{b}$ to belong to Q. Similarly, right-isotonicity, here, differs from the one in [2] in that it does not restrict the path $\mathbf{c}$ to belong to Q.

If the network is modeled by a directed graph as in the TM network model, then the right-monotonicity and right-isotonicity conditions are necessary and sufficient conditions for Dijkstra's algorithm to find the lightest path. However, if the network needs to be modeled as a hypergraph, as is likely the case in AM networks, then these conditions are no longer necessary. Satisfying both conditions is still sufficient for Dijkstra's algorithm to find the lightest path. The modifications introduced to the definitions above do not alter the proof of sufficiency provided in [2] for graphs, and is thus not reproduced here. The lack of necessity is demonstrated next, by presenting an alternative set of sufficient conditions for the optimality of Dijkstra's algorithm.

B. Alternative Dijkstra's sufficient conditions for optimality in hypergraphs

Although right-monotonicity and right-isotonicity conditions are sufficient to show the optimality of Dijkstra's algorithm, they might not be very helpful for path weight functions in AM networks. The right-isotonicity condition, for instance, can only be satisfied if there is a certain decoupling between the nodes in paths $\mathbf{a}$, or $\mathbf{b}$, and those in path $\mathbf{c}$. However, it is precisely the connection between these nodes what we want to include by considering AM networks. In the following, we present a new set of sufficient conditions that guarantee the optimality of Dijkstra's algorithm in directed hypergraphs, with only one hyperedge per node.

Definition 3. *[Condition C1]Consider a route metric* $(Q, \oplus, w, \preceq)$ *defined in a hypergraph with only one hyperedge per node. Given any path* $\mathbf{a}$ *whose last node is A, and the paths* $\mathbf{a} \oplus \langle A,B \rangle$ *and* $\mathbf{a} \oplus \langle A,C \rangle$ *sharing the common root path* $\mathbf{a}$*, we say that the route metric satisfies condition C1 if satisfying any of the conditions below, implies satisfying all the others*

$$w(\mathbf{a} \oplus \langle A,B \rangle) \preceq w(\mathbf{a} \oplus \langle A,C \rangle), \tag{18a}$$
$$w(\mathbf{a} \oplus \langle A,B \rangle) \preceq w(\mathbf{a} \oplus \langle A,B,C \rangle), \tag{18b}$$
$$w(\mathbf{a} \oplus \langle A,B,C \rangle) \preceq w(\mathbf{a} \oplus \langle A,C \rangle), \tag{18c}$$
$$w(\mathbf{a} \oplus \langle A,B \rangle) \preceq w(\mathbf{a} \oplus \langle A,C,B \rangle), \tag{18d}$$
$$w(\mathbf{a} \oplus \langle A,C,B \rangle) \preceq w(\mathbf{a} \oplus \langle A,C \rangle) \tag{18e}$$

for any paths $\mathbf{a} \oplus \langle A,B \rangle$, $\mathbf{a} \oplus \langle A,C \rangle$ *belonging to* Q*. Observe that for hyperpaths with only one edge per node it is also guaranteed that the paths* $\mathbf{a}\oplus\langle A,B,C \rangle$*, and* $\mathbf{a}\oplus\langle A,C,B \rangle$ *belong to* Q.

Definition 4. *[Condition C2]A path weight satisfies condition C2 if for any ordered set of nodes* $\mathbf{c}$*, with partial paths* $\mathbf{c}_{0,j} = \langle \mathbf{c}[0], ..., \mathbf{c}[j] \rangle$, $j = 0, ..., |\mathbf{c}| - 1$ *satisfying* $w(\mathbf{a} \oplus \langle A,B \rangle)) \preceq w(\mathbf{a}\oplus \langle A, \mathbf{c}_{0,j} \rangle)$ *for all* j*, implies that*

$$w(\mathbf{a} \oplus \langle A,B,\mathbf{c} \rangle) \preceq w(\mathbf{a}\oplus \langle A, \mathbf{c} \rangle) \tag{19}$$

for any paths $\mathbf{a} \oplus \langle A,B \rangle$*, and* $\mathbf{a} \oplus \langle A,\mathbf{c} \rangle \in Q$.

Condition C1, basically, implies that given a path $\mathbf{a}\oplus \langle \mathrm{A,C} \rangle$ there exists a lighter path to node C given by $\mathbf{a} \oplus \langle \mathrm{A,B,C} \rangle$, if and only if, the path $\mathbf{a}\oplus \langle \mathrm{A,B} \rangle$ is lighter than $\mathbf{a}\oplus \langle \mathrm{A,C} \rangle$. Condition C2 replaces node C by any complete path $\mathbf{c}$ satisfying $w(\mathbf{a} \oplus \langle \mathrm{A,B} \rangle)) \preceq w(\mathbf{a}\oplus \langle \mathrm{A}, \mathbf{c}_{0,j} \rangle)$ for all j.

The next theorem states the sufficiency of C1 and C2 for the optimality of Dijkstra's algorithm in a directed hypergraph.

Theorem 1. *If a routing metric* $(Q, \oplus, w, \preceq)$ *satisfies C1 and C2, then Dijkstra's algorithm finds the optimal path.*

Proof. Given a hypergraph $\mathcal{H}(V, E)$, the set of nodes $\mathcal{R} = V$, the path weight function w, and the origin node of the path search o, Dijkstra's algorithm returns a set of paths $\mathbf{p}_{\mathrm{o}} = \{\mathbf{p}_{\mathrm{o,r}}, \mathrm{r} \in \mathcal{R}/\mathrm{o}, \}$ from node o to every other network node r$\in \mathcal{R}/\mathrm{o}$, as well as the weights associated to those paths $l_{\mathrm{o}} = \{l_{\mathrm{o,r}}, \mathrm{r} \in \mathcal{R}/\mathrm{o}\}$. The pseudo code of Dijkstra's algorithm is shown in Algorithm 1. Let us denote as $\mathcal{R}^{(i)}$, $\mathbf{p}_{\mathrm{o,r}}^{(i)}$ and $l_{\mathrm{o,r}}^{(i)}$, respectively, the state of the set $\mathcal{R}$ ($\mathcal{R}^{(0)} = V$), the paths $\mathbf{p}_{\mathrm{o,r}}$ and the weights $l_{\mathrm{o,r}}$ at the beginning of the i-th iteration. Suppose the initial iteration is $i = 0$. Let $\mathbf{u}[i]$ be the node extracted from $\mathcal{R}^{(i)}$ in lines 6-7 at iteration i. We say that the path $\mathbf{p}_{\mathrm{o},\mathbf{u}[i]}$ from node o to $\mathbf{u}[i]$ is found at iteration i, as it is no longer updated by the algorithm.

Algorithm 1 Dijkstra's algorithm

$(\boldsymbol{p_o}, \boldsymbol{l_o})$=**Dijkstra**$(\mathcal{R}, w, o)$

```
1:  for each node t∈ R do
2:      l_o,t ← ∞; p_o,t ← NIL
3:  end for
4:  l_o,o ← 1; p_o,o ← o;
5:  while R ≠ ∅ do
6:      u= arg min_{r∈R} l_o,r;
7:      Extract u from R
8:      for each node r∈ R do
9:          compute w_u⊕r = w (p_o,u ⊕ ⟨u,r⟩))
10:         if l_o,r ⪰ w_u⊕r then
11:             l_o,r ← w_u⊕r; p_o,r ← ⟨p_o,u ⊕ ⟨u,r⟩⟩
12:         end if
13:     end for
14: end while
```

We prove this theorem in two steps. First, we show that for a routing metric satisfying C1 in Definition 3, Dijkstra's algorithm at iterations $i = \{1, ..., |V| - 1\}$ finds the path $\mathbf{p}_{\mathrm{o},\mathbf{u}[i]}$ from the origin o to node $\mathbf{u}[i]$ as $\mathbf{u}_{0,i} = \langle \mathbf{u}_{0,i-1}, \mathbf{u}[i] \rangle$ where $\mathbf{u}_{0,i-1}$ is the path found at iteration $i - 1$, ($\mathbf{u}[0]$ =o) and $\mathbf{u}[i]$ is the node that satisfies

$$\mathbf{u}[i] = \arg \min_{\mathrm{r} \in \mathcal{R}^{(i)}} w(\langle \mathbf{u}_{0,i-1}, \mathrm{r} \rangle) \tag{20}$$

where $\mathcal{R}^{(i)} = \mathcal{R}^{(i-1)}/\mathbf{u}_{0,i-1}$. Then, we show that for a routing metric satisfying also C2, the path $\mathbf{u}_{0,i}$ is the lightest path from node o to node $\mathbf{u}[i]$.

<u>Step 1:</u> Suppose $l_{\mathrm{o},r} \succeq w_{\mathrm{u}\oplus\mathrm{r}}$ is satisfied for every node r$\in \mathcal{R}^{(i)}$, from iteration 0 to i. At these iterations, the weights $l_{\mathrm{o,r}}$ and paths $\mathbf{p}_{\mathrm{o,r}}$ are updated $\forall$r$\in \mathcal{R}^{(i)}$ in line 11, as

$$\mathbf{p}_{\mathrm{o,r}}^{(i+1)} = \mathbf{p}_{\mathrm{o},\mathbf{u}[i]}^{(i)} \oplus \langle \mathbf{u}[i], \mathrm{r} \rangle , \tag{21}$$
$$l_{\mathrm{o,r}}^{(i+1)} = w\left(\mathbf{p}_{\mathrm{o,r}}^{(i+1)}\right). \tag{22}$$

At iteration 0, line 6 of Dijkstra's algorithm selects $\mathbf{u}[0]$ =o and thus, the paths $\mathbf{p}_{\mathrm{o,r}}$ $\forall$r$\in \mathcal{R}^{(0)}$ are updated, as $\mathbf{p}_{\mathrm{o,r}}^{(1)} = \langle \mathbf{u}[0], \mathrm{r} \rangle$. At iteration 1, line 6 selects $\mathbf{u}[1]$. Note that $\mathbf{p}_{\mathrm{o},\mathbf{u}[1]}^{(1)} = \langle \mathbf{u}[0], \mathbf{u}[1] \rangle$, and thus, the paths $\forall$r$\in \mathcal{R}^{(1)}$ are updated as $\mathbf{p}_{\mathrm{o,r}}^{(2)} = \langle \mathbf{u}[0], \mathbf{u}[1], \mathrm{r} \rangle$. Observe that at every iteration the paths $\mathbf{p}_{\mathrm{o,r}}$ for every node r, share the common root path $\mathbf{u}_{1,i} = \langle \mathbf{u}[1], ..., \mathbf{u}[i] \rangle$, i.e. $\mathbf{p}_{\mathrm{o,r}}^{(i+1)} = \mathbf{u}_{0,i} \oplus \langle \mathbf{u}[i], \mathrm{r} \rangle = \langle \mathbf{u}_{0,i}, \mathrm{r} \rangle$

for all r. Consequently, node $\mathbf{u}[i]$ in line 6 is chosen according to (20). Observe that

$$\begin{aligned}\mathbf{u}[i] &= \arg\min_{\mathrm{r}\in\mathcal{R}^{(i)}} l_{\mathrm{o,r}}^{(i)}, \\ &= \arg\min_{\mathrm{r}\in\mathcal{R}^{(i)}} w\left(\mathbf{p}_{\mathrm{o,r}}^{(i)}\right), \\ &= \arg\min_{\mathrm{r}\in\mathcal{R}^{(i)}} w(\langle \mathbf{u}_{0,i-1}, \mathrm{r}\rangle). \end{aligned} \tag{23}$$

It only remains to show that $l_{\mathrm{o,r}} \succeq w_{\mathrm{u}\oplus\mathrm{r}}$ in line 10 is always satisfied as previously assumed. To that end, we show that, if $l_{\mathrm{o,r}} \succeq w_{\mathrm{u}\oplus\mathrm{r}}$ is satisfied from iteration 0 to $i-1$, it is also satisfied at iteration i. At the beginning of iteration i, we have that $\mathbf{p}_{\mathrm{o,r}}^{(i)} = \langle \mathbf{u}_{0,i-1}, \mathrm{r}\rangle$ and $l_{\mathrm{o,r}}^{(i)} = w\left(\langle \mathbf{u}_{0,i-1}, \mathrm{r}\rangle\right)$. If node $\mathbf{u}[i]$ is chosen in line 6, according to (20) it is satisfied that $w\left(\langle \mathbf{u}_{0,i-1}, \mathbf{u}[i]\rangle\right) \preceq w\left(\langle \mathbf{u}_{0,i-1}, \mathrm{r}\rangle\right)$ for all r$\in \mathcal{R}^{(i)}$, then, in line 9, we compute

$$w_{\mathrm{u}\oplus\mathrm{r}} = w\left(\langle \mathbf{u}_{0,i-1}, \mathbf{u}[i], \mathrm{r}\rangle\right). \tag{24}$$

It is then a direct consequence of C1, that

$$\begin{aligned} w_{\mathrm{u}\oplus\mathrm{r}} &= w\left(\langle \mathbf{u}_{0,i-1}, \mathbf{u}[i], \mathrm{r}\rangle\right), \\ &\preceq w\left(\langle \mathbf{u}_{0,i-1}, \mathrm{r}\rangle\right), \\ &= l_{\mathrm{o,r}}^{(i)} \end{aligned} \tag{25}$$

and thus, $l_{\mathrm{o,r}} \succeq w_{\mathrm{u}\oplus\mathrm{r}}$ is also satisfied at the $i-$th iteration for every node. Finally, notice that the $l_{\mathrm{o,r}} \succeq w_{\mathrm{u}\oplus\mathrm{r}}$ is trivially satisfied at $i=0$, since initially $l_{\mathrm{o,r}} = \infty$ for all r.

Step 2: Next, we show that if a routing metric satisfies C1 and C2, then the path $\mathbf{u}_{0,i}$ is the lightest paths from node $\mathbf{u}[0]$ to node $\mathbf{u}[i]$. We prove this result by contradiction. Assume that the lightest path $\mathbf{s}$ between a given source-destination pair S-D satisfies $\mathbf{s}[i] = \mathbf{u}[i]$ for $j < i$ but $\mathbf{s}[i] \neq \mathbf{u}[i]$. Given that $\mathbf{u}[i]$ is choose according to (20), we have that

$$w\left(\langle \mathbf{s}_{0,i-1}, \mathbf{u}[i]\rangle\right) \preceq w\left(\langle \mathbf{s}_{0,i-1}, \mathbf{s}[i]\rangle\right).$$

Let us denote B=$\mathbf{u}[i]$. We define a new path from the source S to the destination D by including node B between nodes $\mathbf{s}[i-1]$ and $\mathbf{s}[i]$, namely $\langle \mathbf{s}_{0,i-1}, \mathrm{B}, \mathbf{s}_{i,L+1}\rangle$. We show next that this path is lighter than the original path $\mathbf{s} = \langle \mathbf{s}_{0,i-1}, \mathbf{s}_{i,L+1}\rangle$ which contradicts the assumption that $\mathbf{s}$ is the lightest path. Denote $\mathbf{a} = \mathbf{s}_{0,i-1}$, and $\mathbf{c} = \mathbf{s}_{i,L+1}$ with $\mathbf{c}[0] = \mathbf{s}[i]$ =C. In this case, the newly defined path reads $\langle \mathbf{a}, \mathrm{B}, \mathbf{c}\rangle$, and from (20), we have that

$$w(\langle \mathbf{a}, \mathrm{B}\rangle) \preceq w\left(\langle \mathbf{a}, \mathrm{C}\rangle\right). \tag{26}$$

We consider the following two possible situations: *i*) Assume that the partial path $\mathbf{c}$ satisfies $w\left(\langle \mathbf{a}, \mathrm{B}\rangle)\right) \preceq w(\langle \mathbf{a}, \mathbf{c}_{0,j}\rangle)$ for all j. In this case, it is a direct consequence of C2 that $w(\langle \mathbf{a}, \mathrm{B}, \mathbf{c}\rangle) \preceq w(\langle \mathbf{a}, \mathbf{c}\rangle)$ or, equivalently, that $w(\langle \mathbf{s}_{0,i-1}, \mathrm{B}, \mathbf{s}_{i,L+1}\rangle) \preceq w(\mathbf{s})$. *ii*) Instead, assume that the partial path $\mathbf{c}$ satisfies

$$w\left(\langle \mathbf{a}, \mathrm{B}\rangle)\right) \preceq w(\langle \mathbf{a}, \mathbf{c}_{0,j}\rangle), \tag{27}$$

for all $j < m$ but not at $j = m$, this is

$$w\left(\langle \mathbf{a}, \mathrm{B}\rangle\right) \succeq w(\langle \mathbf{a}, \mathbf{c}_{0,m-1}, \mathrm{D}\rangle) \tag{28}$$

with $\mathbf{c}[m]$ =D. In this case, we show next, that C1 and C2 require that $w\left(\langle \mathbf{a}, \mathrm{B}\rangle)\right) \succeq w(\langle \mathbf{a}, \mathrm{D}\rangle)$, which contradicts the assumption that node B is chosen in (20). To prove this result, we iteratively remove the node prefixed to node D in the path $\langle \mathbf{a}, \mathrm{B}, \mathbf{c}_{0,m-1}, \mathrm{D}\rangle$, until node B is finally removed. We show that a lighter path to node D is obtained at every iteration. We begin by removing node $\mathbf{c}[m-1]$. Denote, $\mathbf{c}[m-1]$=C and $\mathbf{a}' = \langle \mathbf{a}, \mathrm{B}, \mathbf{c}_{0,m-2}\rangle$. Then, combining (27) for $j = m-1$, and (28), we have that

$$\begin{aligned} w(\langle \mathbf{a}', \mathrm{C}, \mathrm{D}\rangle) &\preceq w\left(\langle \mathbf{a}, \mathrm{B}\rangle\right), & \text{(29a)} \\ &\preceq w(\langle \mathbf{a}', \mathrm{C}\rangle) & \text{(29b)} \end{aligned}$$

and thus, satisfying C1 implies that

$$w(\langle \mathbf{a}', \mathrm{D}\rangle) \preceq w(\langle \mathbf{a}', \mathrm{C,D}\rangle) \tag{30}$$

or, equivalently,

$$w(\langle \mathbf{a}, \mathrm{B}, \mathbf{c}_{0,m-2}, \mathrm{D}\rangle) \preceq w(\langle \mathbf{a}, \mathrm{B}, \mathbf{c}_{0,m-1,}, \mathrm{D}\rangle). \tag{31}$$

This is, by removing node $\mathbf{c}[m-1]$, we obtain a lighter path to node D. Next, we remove node $\mathbf{c}[m-2]$. By combining (27) with (28) for $j = m-2$, we can write

$$\begin{aligned} w(\langle \mathbf{a}, \mathrm{B}, \mathbf{c}_{0,m-3}, \mathbf{c}[m-2]\rangle) &\succeq w(\langle \mathbf{a}, \mathrm{B}, \mathbf{c}_{0,m-1}, \mathrm{D}\rangle), & \text{(32a)} \\ &\succeq w(\langle \mathbf{a}, \mathrm{B}, \mathbf{c}_{0,m-2,} \mathrm{D}\rangle). & \text{(32b)} \end{aligned}$$

Now, denote C= $\mathbf{c}\,[m-2]$ and $\mathbf{a}' = \langle \mathbf{a}, \mathrm{B}, \mathbf{c}_{0,m-3}\rangle$, then (32b) can be rewritten as

$$w(\langle \mathbf{a}', \mathrm{C}, \mathrm{D}\rangle) \preceq w(\langle \mathbf{a}', \mathrm{C}\rangle) \tag{33}$$

and thus, following previous arguments, satisfying C1 implies

$$\begin{aligned} w(\langle \mathbf{a}, \mathrm{B}, \mathbf{c}_{0,m-3,} \mathrm{D}\rangle) &\preceq w(\langle \mathbf{a}, \mathrm{B}, \mathbf{c}_{0,m-2,}, \mathrm{D}\rangle), & \text{(34a)} \\ &\preceq w(\langle \mathbf{a}, \mathrm{B}, \mathbf{c}_{0,m-1,}, \mathrm{D}\rangle). & \text{(34b)} \end{aligned}$$

This is, by removing node $\mathbf{c}[m-2]$, we obtain a lighter path to node D. We repeat this procedure until node B is removed. □

V. Optimality Analysis of Minimum Energy Accumulative Routing Metrics

In this section, we study the optimality of Dijkstra's algorithm for the minimum energy path weight functions encountered in Section III.

A. Graphs

We begin by discussing the optimality of Dijkstra's algorithm for the minimum energy path weight functions that can be evaluated over a graph. In that case, optimality is guaranteed by showing that the route metric satisfies right-monotonocity and right-isotonocity [2]. This includes the TM network, the DF SAM network with a fixed first relay node, the DF DAM network with a fixed last relay node, and the PF DAM network. For simplicity, in this section, we denote as $\mathbf{p}$, both, the forward and the backward paths. Then, for the TM, DF SAM and DF DAM networks, the path weight function admits the following recursive function computation

$$w(\mathbf{p}_i) = \begin{cases} \frac{1}{\beta_{\mathbf{p}[i-1],\mathbf{p}[i]}} + w(\mathbf{p}_{i-1}), & \text{if } w(\mathbf{p}_i) \geq w(\mathbf{p}_{i-1}) \\ \infty & \text{otherwise} \end{cases} \tag{35}$$

where $\beta_{\text{u,v}}$ is the metric associated to the edge $e_{\text{u,v}}$. For TM $\beta_{\text{u,v}} = g_{\text{u,v}}$, for DF SAM $\beta_{\text{u,v}} = \frac{g_{\text{u,v}} g_{\text{S,R}}}{g_{\text{S,R}} - g_{\text{S,v}}}$, and for DF DAM $\beta(e_{\text{u,v}}) = \beta_{\text{u,v}} = \frac{h_{\text{u,v}} h_{\text{D,R}}}{h_{\text{D,R}} - h_{\text{D,v}}}$.

Right-monotonicity is implicit in the path weight definition (35) as we require $w(\mathbf{p}_i) \geq w(\mathbf{p}_{i-1})$. This condition is always satisfied for the TM network as $g_{\text{u,v}} \geq 0$ for all u and v. For the DF SAM network with the first relay R fixed, this condition implies that only nodes v with a better channel from the source than that from the source to the first relay $g_{\text{S,R}} \geq g_{\text{S,v}}$ are eligible for the optimal path. Similarly, for the DF DAM network with the last relay R fixed, this condition implies that only nodes with a better channel to the destination than that of the last relay are eligible.

To show right-isotonicity, consider the paths $\mathbf{a}$ and $\mathbf{b}$ between nodes A and B, with $w(\mathbf{a})$ and $w(\mathbf{b})$ satisfying $w(\mathbf{a}) \leq w(\mathbf{b})$. Let us concatenate node C to the right of $\mathbf{a}$ and $\mathbf{b}$, then the path weight at node C is given by

$$w(\mathbf{a} \oplus \langle \text{B,C} \rangle) = \frac{1}{\beta_{\text{B,C}}} + w(\mathbf{a} \oplus \text{C}), \tag{36a}$$

$$w(\mathbf{b} \oplus \langle \text{B,C} \rangle) = \frac{1}{\beta_{\text{B,C}}} + w(\mathbf{b} \oplus \text{C}). \tag{36b}$$

Given that $\beta_{\text{B,C}} \geq 0$ then $w(\mathbf{a}) \leq w(\mathbf{b})$ implies $w(\mathbf{a} \oplus \langle \text{B,C} \rangle) \leq w(\mathbf{b} \oplus \langle \text{B,C} \rangle)$.

For the DF SAM these optimality results allow us to find optimal paths from a source-relay pair $\langle$S,R$\rangle$ to every other network node using Dijkstra's algorithm. We however need to repeat the search for all possible first relay nodes in order to get the optimal path from the source node. Similarly, for the DF DAM network, we need to repeat the search for all possible last relays nodes.

For the PF DAM network, the metrics associated to the edge $e_{\text{u,v}}$ is given by $\beta(e_{\text{u,v}}) = \{h_{\text{u,v}}, h_{\text{D,v}}\} = \{g_{\text{v,u}}, g_{\text{v,D}}\}$ and the path weight function reads

$$w(\mathbf{p}_i) = \frac{1}{h_{\mathbf{p}[i-1],\mathbf{p}[i]}} + \left(1 - \frac{h_{\text{D},\mathbf{p}[i]}}{h_{\mathbf{p}[i-1],\mathbf{p}[i]}}\right) w(\mathbf{p}_{i-1}) \tag{37}$$

if $h_{\mathbf{p}[i-1],\mathbf{p}[i]} \geq h_{\text{D},\mathbf{p}[i]}$ and $w(\mathbf{p}_i) = \infty$ otherwise. Right-monotonicity is satisfied since $h_{\mathbf{p}[i-1],\mathbf{p}[i]} > 0$ and $h_{\mathbf{p}[i-1],\mathbf{p}[i]} \geq h_{\mathbf{p}[0],\mathbf{p}[i]}$ together imply $w(\mathbf{p}_i) \geq w(\mathbf{p}_{i-1})$. To show right-isotonocity, observe that the path weight at node C is given by

$$w(\mathbf{a} \oplus \langle \text{B,C} \rangle) = \frac{1}{h_{\text{B,C}}} + \left(1 - \frac{h_{\text{D,C}}}{h_{\text{B,C}}}\right) w(\mathbf{a} \oplus \text{C}), \tag{38a}$$

$$w(\mathbf{b} \oplus \langle \text{B,C} \rangle) = \frac{1}{h_{\text{B,C}}} + \left(1 - \frac{h_{\text{D,C}}}{h_{\text{B,C}}}\right) w(\mathbf{b} \oplus \text{C}). \tag{38b}$$

Given that $h_{\text{B,C}} \geq 0$ and $1 - \frac{h_{\text{D,C}}}{h_{\text{B,C}}} \geq 0$, then $w(\mathbf{a}) \leq w(\mathbf{b})$ implies $w(\mathbf{a} \oplus \langle \text{B,C} \rangle) \leq w(\mathbf{b} \oplus \langle \text{B,C} \rangle)$.

B. Hypergraphs

We have shown that for the general DF AM network, the CB AM network, and the DF EAM network, the minimum energy path weight function need to be computed over a hypergraph. It is well-know that the problem of finding the optimal minimum energy paths for the DF AM network is NP-complete [3] and [4]. Indeed, it can be shown that, although the DF AM minimum energy path weight satisfies right-monotonicity, it does not satisfy right-isotonicity, nor C1 and C2, and thus, the optimality of Dijkstra's algorithm can not be guaranteed.

For the CB and DF EAM, the path weight function can be expressed in a unify manner as

$$w(\mathbf{p}_i) = \frac{1 + \sum_{j=1}^{i-1} \beta_{j,i} w(\mathbf{p}_j)}{\sum_{j=0}^{i-1} \beta_{j,i}} \tag{39}$$

with $\beta_{j,i} = h_{j,i}$ for CB and $\beta_{j,i} = m_{j,i} = g_{j,i} - g_{j-1,i}$ for DF EAM. The path weight function (39) neither satisfies right-monotonicity nor right-isotonicity. However, as we show in the following theorem it satisfies C1 and C2 if $\beta_{\text{u,v}} \geq 0$ and thus, the optimal path can be found using Dijkstra's algorithm.

Theorem 2. *Consider a hypergraph $\mathcal{H}(V, E)$ with only one edge per node, where the metric associated to the hyperedge of node u is defined as $\beta(e_u) = \{\beta_{u,v}, \forall v\}$. The path weight function (39) satisfies C1 and C2 if $\beta_{u,v} \geq 0$.*

Proof. We first show that C1 is satisfied. Let $\mathbf{a}$ be a path with L_a relays nodes, $\mathbf{a} = \langle \mathbf{a}[0], ..., \mathbf{a}[L_a + 1] \rangle$. Given the paths $\langle \mathbf{a}$,X$\rangle$, where X is any network node X$\in V$, define

$$N_{\text{X}} = 1 + \sum_{l=1}^{L_{\mathbf{a}}+1} \beta_{\mathbf{a}[l],\text{X}} w(\mathbf{a}_{0,j}),$$

$$D_{\text{X}} = \sum_{l=0}^{L_a+1} \beta_{\mathbf{a}[l],\text{X}}.$$

Then, the weight of the paths evaluated in condition C1 are given by

$$w(\langle \mathbf{a}, \text{C} \rangle) = \frac{N_{\text{C}}}{D_{\text{C}}}, \tag{40}$$

$$w(\langle \mathbf{a}, \text{B} \rangle) = \frac{N_{\text{B}}}{D_{\text{B}}}, \tag{41}$$

$$w(\langle \mathbf{a}, \text{B}, \text{C} \rangle) = \frac{N_{\text{C}} + \beta_{\text{B,C}} w(\langle \mathbf{a}, \text{B} \rangle)}{D_{\text{C}} + \beta_{\text{B,C}}}, \tag{42}$$

$$w(\langle \mathbf{a}, \text{C}, \text{B} \rangle) = \frac{N_{\text{B}} + \beta_{\text{C,B}} w(\langle \mathbf{a}, \text{C} \rangle)}{D_{\text{B}} + \beta_{\text{C,B}}}. \tag{43}$$

Suppose that $w(\langle \mathbf{a}, \text{B} \rangle) \preceq w(\langle \mathbf{a}, \text{C} \rangle)$, and $\beta_{\text{B,C}}, \beta_{\text{C,B}} \geq 0$, then observe that

$$\begin{aligned} w(\langle \mathbf{a}, \text{B}, \text{C} \rangle) &= \frac{N_{\text{C}} + \beta_{\text{B,C}} w(\langle \mathbf{a}, \text{B} \rangle)}{D_{\text{C}} + \beta_{\text{B,C}}}, \\ &\preceq \frac{N_{\text{C}} + \beta_{\text{B,C}} w(\langle \mathbf{a}, \text{C} \rangle)}{D_{\text{C}} + \beta_{\text{B,C}}}, \\ &= w(\langle \mathbf{a}, \text{C} \rangle) \end{aligned} \tag{44}$$

and

$$\begin{aligned} w(\langle \mathbf{a}, \text{C}, \text{B} \rangle) &= \frac{N_{\text{B}} + \beta_{\text{C,B}} w(\langle \mathbf{a}, \text{C} \rangle)}{D_{\text{B}} + \beta_{\text{C,B}}}, \\ &\succeq \frac{N_{\text{B}} + \beta_{\text{C,B}} w(\langle \mathbf{a}, \text{B} \rangle)}{D_{\text{B}} + \beta_{\text{C,B}}}, \\ &= w(\langle \mathbf{a}, \text{B} \rangle). \end{aligned} \tag{45}$$

By isolating $w(\langle \mathbf{a}, \mathrm{B}\rangle)$ from (42) and using (44), we observe that

$$\begin{aligned} w(\langle \mathbf{a}, \mathrm{B}\rangle) &= w(\langle \mathbf{a}, \mathrm{B}, \mathrm{C}\rangle) + \frac{D_{\mathrm{C}} w(\langle \mathbf{a}, \mathrm{B}, \mathrm{C}\rangle) - N_{\mathrm{C}}}{\beta_{\mathrm{B,C}}}, \\ &\preceq w(\langle \mathbf{a}, \mathrm{B}, \mathrm{C}\rangle) + \frac{D_{\mathrm{C}} w(\langle \mathbf{a}, \mathrm{C}\rangle) - N_{\mathrm{C}}}{\beta_{\mathrm{B,C}}}, \\ &= w(\langle \mathbf{a}, \mathrm{B}, \mathrm{C}\rangle). \end{aligned} \tag{46}$$

Notice that, the reverse is also true, i.e. satisfying (44) implies

$$\begin{aligned} w(\langle \mathbf{a}, \mathrm{B}\rangle) &= \frac{w(\langle \mathbf{a}, \mathrm{B}, \mathrm{C}\rangle)(D_{\mathrm{C}} + \beta_{\mathrm{B,C}}) - N_{\mathrm{C}}}{\beta_{\mathrm{B,C}}}, \\ &\preceq \frac{w(\langle \mathbf{a}, \mathrm{C}\rangle)(D_{\mathrm{C}} + \beta_{\mathrm{B,C}}) - N_{\mathrm{C}}}{\beta_{\mathrm{B,C}}}, \\ &= w(\langle \mathbf{a}, \mathrm{C}\rangle). \end{aligned} \tag{47}$$

Next, by isolating $w(\langle \mathbf{a}, \mathrm{C}\rangle)$ from (43) and using (45), we observe that

$$\begin{aligned} w(\langle \mathbf{a}, \mathrm{C}\rangle) &= w(\langle \mathbf{a}, \mathrm{C}, \mathrm{B}\rangle) + \frac{w\langle \mathbf{a}, \mathrm{C}, \mathrm{B}\rangle) D_{\mathrm{B}} - N_{\mathrm{B}}}{\beta_{\mathrm{C,B}}}, \\ &\succeq w(\langle \mathbf{a}, \mathrm{C}, \mathrm{B}\rangle) + \frac{w(\langle \mathbf{a}, \mathrm{B}\rangle) D_{\mathrm{B}} - N_{\mathrm{B}}}{\beta_{\mathrm{C,B}}}, \\ &= w(\langle \mathbf{a}, \mathrm{C}, \mathrm{B}\rangle). \end{aligned} \tag{48}$$

Next, we show that C2 is also satisfied. Let us consider the paths $\langle \mathbf{a}, \mathbf{c}\rangle$ and $\langle \mathbf{a}, \mathrm{B}, \mathbf{c}\rangle$ where $\mathbf{c} = \langle \mathbf{c}[0], ..., \mathbf{c}[L_{\mathbf{c}} + 1]\rangle$ and denote the weight of the paths $\langle \mathbf{a}, \mathbf{c}_{0,i}\rangle$ as

$$w(\langle \mathbf{a}, \mathbf{c}_{0,i}\rangle) = \frac{N_i}{D_i} \tag{49}$$

for $i = 0, ..., L_{\mathbf{c}} + 1$. Let us decompose N_i and D_i as $N_i = N_{\mathbf{a},i} + N_{\mathbf{c},i}$ and $D_i = D_{\mathbf{a},i} + D_{\mathbf{c},i}$ where

$$N_{\mathbf{a},i} = 1 + \sum_{l=0}^{L_{\mathbf{a}}+1} \beta_{\mathbf{a}[l],\mathbf{c}[i]} w(\mathbf{a}_{0,l}),\ D_{\mathbf{a},i} = \sum_{l=0}^{L_a+1} \beta_{\mathbf{a}[l],\mathbf{c}[i]},$$

$$N_{\mathbf{c},i} = \sum_{l=0}^{i-1} \beta_{\mathbf{c}[l],\mathbf{c}[i]} w(\langle \mathbf{a}, \mathbf{c}_{0,l}\rangle),\ D_{\mathbf{c},i} = \sum_{l=0}^{i-1} \beta_{\mathbf{a}[l],\mathbf{c}[i]}.$$

Then, observe that if $w(\langle \mathbf{a}, \mathrm{B}\rangle) \preceq w(\langle \mathbf{a}, \mathbf{c}_{0,l}\rangle)$ for all l and $w(\langle \mathbf{a}, \mathrm{B}, \mathbf{c}_{0,l}\rangle) \preceq w(\langle \mathbf{a}, \mathbf{c}_{0,l}\rangle)$ for $l = 1$ to $l = i - 1$, then $w(\langle \mathbf{a}, \mathrm{B}, \mathbf{c}_{0,i}\rangle)$

$$= \frac{N_{\mathbf{a},i} + \beta_{\mathrm{B},\mathbf{c}[i]} w(\langle \mathbf{a}, \mathrm{B}\rangle) + \sum_{l=0}^{i-1} \beta_{\mathbf{c}[l],\mathbf{c}[i]} w(\langle \mathbf{a}, \mathrm{B}, \mathbf{c}_{0,l}\rangle)}{D_{\mathbf{a},i} + \beta_{\mathrm{B},\mathbf{c}[i]} + D_{\mathbf{c},i}}, \tag{50a}$$

$$\preceq \frac{N_i + \beta_{\mathrm{B},\mathbf{c}[i]} w(\langle \mathbf{a}, \mathrm{B}\rangle)}{D_i + \beta_{\mathrm{B},\mathbf{c}[i]}}. \tag{50b}$$

$$= \frac{N_i + \beta_{\mathrm{B},\mathbf{c}[i]} \frac{w(\langle \mathbf{a}, \mathrm{B}\rangle)}{w(\mathbf{a}, \mathbf{c}_{0,i})} \frac{N_i}{D_i}}{D_i + \beta_{\mathrm{B},\mathbf{c}[i]}}, \tag{50c}$$

$$\preceq \frac{N_i + \beta_{\mathrm{B},\mathbf{c}[i]} \frac{N_i}{D_i}}{D_i + \beta_{\mathrm{B},\mathbf{c}[i]}}, \tag{50d}$$

$$= \frac{N_i}{D_i} = w(\langle \mathbf{a}, \mathbf{c}_{0,i}\rangle). \tag{50e}$$

where inequality (50b) is due to $w(\langle \mathbf{a}, \mathrm{B}, \mathbf{c}_{0,l}\rangle) \preceq w(\langle \mathbf{a}, \mathbf{c}_{0,l}\rangle)$ for $l = 0, ..., i - 1$ and inequality (50d) is due to $w(\langle \mathbf{a}, \mathrm{B}\rangle) \preceq w(\mathbf{a}, \mathbf{c}_{0,i})$. $\square$

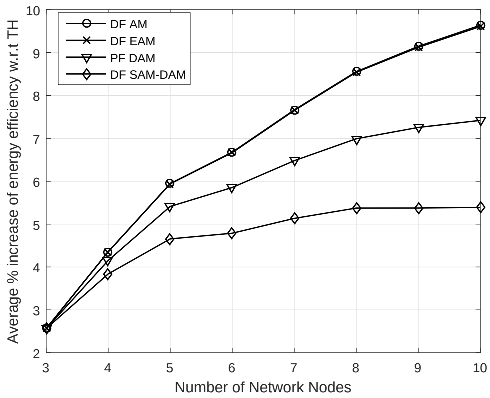

Fig. 12. Average %IoE as a function of the number of network nodes ($\rho = 1$, $\nu = 3$).

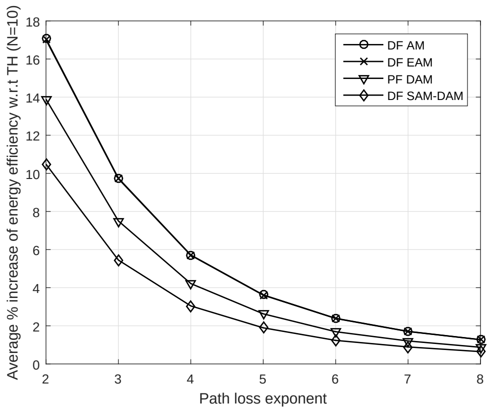

Fig. 13. Average %IoE as a function of the path loss exponent ($N = 10$, $\rho = 1$).

VI. Numerical Results

The numerical results presented here aim first at showing the benefits of accumulative multi-hoping versus traditional multi-hoping for the simplistic communication model considered in Section III and second, at evaluating the performance of the proposed DF EAM heuristic routing protocols for the DF AM network. We provide results for the commonly used geometric propagation channel model: the channel gain between nodes i and j is $g_{i,j} = d_{i,j}^{-\nu}$ where d_{ij} is the distance between these two nodes and ν is the path-loss exponent. For each simulation case, we generate $M = 1000$ independent scenarios with N nodes randomly and uniformly placed in a square plane of area $A = \frac{N}{\rho}$, where ρ [#nodes/m^2] stands for the density of nodes. For each scenario, we compute the path weights from every network node $i = [1, ...N]$ to every node $j \neq i$, assuming that the rest of nodes are potential relays. Unless stated otherwise, we use $N = 10$, $\rho = 1$ and $\nu = 3$.

Recall that the path weight functions in Section III are inversely proportional to the energy efficiency of a path. This is, lower weights means more efficient paths. Given a source destination pair let, w^{TH} denote the TH weight of the optimal TH path and w^{AM} denote the weight of the optimal path for any of AM strategies discussed in Section III, namely DF AM, DF SAM, DF DAM, PF DAM or DF EAM. Then, we measure the percentage increase of energy efficiency (IoE) achieved by the AM strategies with respect to the TM strategy

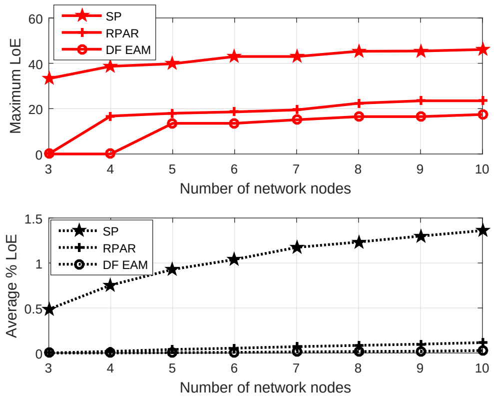

Fig. 14. Average (dotted) and maximum (solid) %LoE as a function of the number of network nodes ($\rho = 1$, $\nu = 3$).

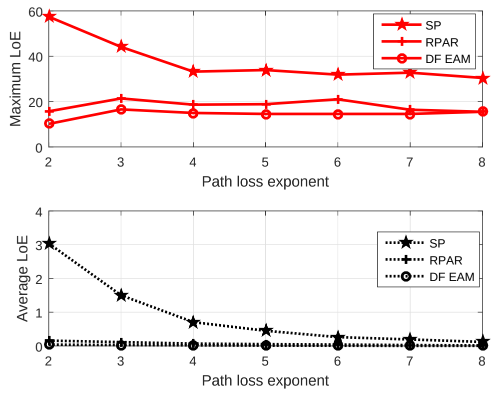

Fig. 15. Average (dotted) and maximum (solid) %LoE as a function of the path loss exponent ($N = 10$, $\rho = 1$).

as IoE $= 100\frac{w^{TM}-w^{AM}}{w^{AM}}$. Notice that due to symmetry, for the DF DAM and DF SAM networks we always obtain the same average results and thus, we consider them together.

Before showing the numerical results it is meaningful to discuss the path search algorithm complexity in each case. Finding the optimal path for the DF AM network by exhaustive search has complexity order $O\left((N-1)!\right)$, see [8]. It is well known that Dijkstra's algorithm has complexity order $O(N^2)$. Thus, it follows that finding the optimal path for TH and PF DAM networks has complexity order $O(N^2)$. For the DF SAM and DF SAM networks, complexity order is $O(N^3)$ as we need to repeat the search for every possible first and last relay, respectively. Finally, the complexity order of the DF EAM is $O(N^4)$ since the extended hypergraph has $O\left(N^2\right)$ nodes. In Fig. 12 we depict the average IoE for each accumulative strategy as a function of the number of network nodes. Although we consider fixed values for the density of nodes $\rho = 1$, the same behavior is observed in any other configuration. In general, the accumulative gain increases with the number of network nodes. Observe that DF EAM and DF AM achieve very similar performance. In Fig. 13 we depict the average IoE for each accumulative strategy as a function of the path loss exponent for a network with $N = 10$ nodes. As the path loss exponent increases the gains of accumulative strategies decrease. However, for common path loss exponents 3-5 there exist substantial gains through energy/information accumulation.

In Figures 12 and 13, we have observed that the EAM algorithm achieves a IoE similar to that of the exhaustive search in a DF AM network. Next, we discuss this result in more detail. The DF EAM algorithm can be seen as a low complexity heuristic routing solution for the DF AM network. Other heuristic algorithm for the DF AM network have been proposed in the literature. The SP algorithm proposed in [4], also referred to as CTNCR in [8], and as Heuristic 1 in [15] consists of finding the lightest path for the DF TM network with Dijkstra's algorithm and then using it to communicate in the DF AM network. Also in [4], authors proposed the RPAR algorithm, also refereed to as SNER in [8]. In RPAR, Dijkstra's algorithm is, directly, applied to the DF AM path weight function in (7) to find good paths. As we discuss in [13], the performance of these algorithms can be improved by repeating the search forward and backward and then choosing the best path candidate. Given a source destination pair let us denote as w^{DFAM} the weight of the optimal path in the DF AM network and as w^{H} the weight of the path found with each of of the heuristic algorithms considered, namely, DF EAM, SP and RPAR. Then, we define the loss of efficiency (LoE) of the heuristic algorithm with respect the optimal DF AM path as LoE $= 100\frac{w^{H}-w^{DFAM}}{w^{DFAM}}$. In Fig. 14, we depict the average and maximum LoE for each of the three heuristic algorithms considered as a function of the number of network nodes, after 1000 randomly generated scenarios. We observe that on average the LoE is quite low for all three heuristic algorithms, specially for RPAR and DF EAM. Thus, we focus on discussing the worst case results (maximum LoE curves). We observe that the RPAR algorithm for $N \leq 3$, and the DF EAM algorithm for $N \leq 4$, always find the optimal path. This is so, as it can be shown that in those cases, these algorithms coincide with the exhaustive search. Observe that the proposed DF EAM algorithm improves the RPAR algorithm by more than 5% and the SP algorithm by more than 25% for networks with more than 5 nodes. This is achieved at the cost of additional complexity. For the SP and SNER algorithms, we only need to run the Dijkstra's algorithm and thus, the complexity order is $O(N^2)$, as discussed above. Recall that the complexity of the EAM algorithm is $O(N^4)$. Finally, in Fig. 15 we depict the average and maximum LoE for each the three heuristic algorithms considered as a function of the path loss exponent for a network with $N = 10$ nodes. Focusing, again, on the worst case results, we observe that for the SP algorithm the LoE decreases with the path loss exponent, whereas for the other two heuristic algorithms the LoE remains constant for all the range of path loss exponents evaluated.

VII. Conclusions

In this paper, we studied the routing problem in accumulative multi-hop networks. We showed that as opposed to traditional multi-hoping where the network is well modeled by a graph, for routing in accumulative networks, the network needs to be modeled by a hypergraph. We studied the properties that guarantee that Dijkstra's algorithm finds the optimal path in such networks, and presented sufficient conditions for the optimality. These conditions are particularized for the minimum energy routing problem with decode-and-forward relays, parity-forwarding relays, and for the cut-set bound.

Appendix

In this appendix, we obtain the backward path weigh function in (8). We begin by rewriting (7) as

$$0=\sum_{j=0}^{i-1} g_{j,i}\Delta_j\left(\mathbf{p}\right)-1 \tag{51}$$

where $\Delta_j\left(\mathbf{p}\right)\triangleq w(\mathbf{p}_{j+1})-w(\mathbf{p}_j)$, $j=0,...,L$, and $w(\mathbf{p}_0)=0$. Observe that $w(\mathbf{p})=\sum_{j=0}^{L}\Delta_j(\mathbf{p})$. Similarly, we write (8) as

$$0 \;=\; \sum_{j=0}^{i-1} h_{j,i}\Delta_j(\overleftarrow{\mathbf{p}})-1. \tag{52}$$

We need to show that if $h_{i,j}=g_{L+1-j,L+1-i}$ then $w(\mathbf{p}_{L+1})=w(\overleftarrow{\mathbf{p}}_{L+1})$. Observe that

$$\begin{aligned}
0 &= \sum_{j=0}^{L} h_{j,L+1}\Delta_j\left(\overleftarrow{\mathbf{p}}\right)-1, && (53)\\
&= \sum_{j=0}^{L} g_{0,L+1-j}\Delta_j\left(\overleftarrow{\mathbf{p}}\right)-1, && (54)\\
&= \sum_{j=0}^{L}\left(1-\sum_{l=1}^{L-j} g_{l,L+1-j}\Delta_l\left(\mathbf{p}\right)\right)\Delta_j\left(\overleftarrow{\mathbf{p}}\right)-\Delta_0\left(\overleftarrow{\mathbf{p}}\right), && (55)\\
&= w(\overleftarrow{\mathbf{p}})-\sum_{j=0}^{L}\Delta_j\left(\overleftarrow{\mathbf{p}}\right)\sum_{l=0}^{L-j} g_{l,L+1-j}\Delta_l\left(\mathbf{p}\right), && (56)\\
&= w(\overleftarrow{\mathbf{p}})-\sum_{l=0}^{L}\Delta_l\left(\mathbf{p}\right)\sum_{j=0}^{L-l} g_{l,L+1-j}\Delta_j\left(\overleftarrow{\mathbf{p}}\right), && (57)\\
&= w(\overleftarrow{\mathbf{p}})-\sum_{m=1}^{L+1}\Delta_{L+1-m}\left(\mathbf{p}\right)\sum_{j=0}^{m-1} g_{L+1-l,L+1-j}\Delta_j\left(\overleftarrow{\mathbf{p}}\right), && (58)\\
&= w(\overleftarrow{\mathbf{p}})-w(\mathbf{p}). && (59)
\end{aligned}$$

Equality (53) follows from (52) for $i=L+1$. Equality (54) follows from channel reciprocity $h_{j,i}=g_{L+1-i,L+1-j}$. Equality (55) follows from isolating $g_{0,L+1-j}$ in its (51). To get equality (56), we use $w(\overleftarrow{\mathbf{p}})=\sum_{j=0}^{L}\Delta_j(\overleftarrow{\mathbf{p}})$ and $\sum_{j=0}^{L} g_{0,L+1-j}\Delta_j(\overleftarrow{\mathbf{p}})=1$. Then, equality (57) is obtained by reordering the elements in the sums. In (58), we define $m=L+1-l$. Finally, we get (59) by using $\sum_{j=0}^{m-1} g_{L+1-l,L+1-j}\Delta_j(\overleftarrow{\mathbf{p}}))=1$ and $w(\mathbf{p})=\sum_{m=1}^{L+1}\Delta_{L+1-m}(\mathbf{p})$.

References

[1] J. Sobrinho, "An algebraic theory of dynamic network routing," *Networking, IEEE/ACM Transactions on*, vol. 13, no. 5, pp. 1160 – 1173, oct. 2005.

[2] Y. Yang and J. Wang, "Design guidelines for routing metrics in multihop wireless networks," in *IEEE INFOCOM*, 2008, pp. 1615 –1623.

[3] I. Maric and R. Yates, "Cooperative multihop broadcast for wireless networks," *IEEE J. Select. Areas Commun.*, vol. 22, no. 6, pp. 1080–1088, Aug. 2004.

[4] J. Chen, L. Jia, X. Liu, G. Noubir, and R. Sundaram, "Minimum energy accumulative routing in wireless networks," in *Proc. IEEE INFOCOM*, vol. 3, Mar. 2005, pp. 1875 – 1886.

[5] A. Molisch, N. Mehta, J. Yedidia, and J. Zhang, "Cooperative relay networks using fountain codes," in *Proc. IEEE Global Communications Conference (GLOBECOM)*, Nov. 2006.

[6] J. Castura and Y. Mao, "Rateless coding over fading channels," *Communications Letters, IEEE*, vol. 10, no. 1, pp. 46–48, Jan 2006.

[7] S. C. Draper, L. Liu, A. F. Molisch, and J. S.Yedidia, "Cooperative routing for wireless networks using mutual-information accumulation," *IEEE Trans. Inform. Theory*, vol. 57, Aug. 2011.

[8] Z. Yang and A. Høst-Madsen, "Routing and power allocation in asynchronous gaussian multiple-relay channels," *EURASIP Journal on Wireless Communications and Networking*, 2006.

[9] T. Girici and A. C. Kazez, "Energy efficient routing with mutual information accumulation," in *Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt), 2012 10th International Symposium on*, may 2012, pp. 425 –430.

[10] R. Yim, N. Mehta, A. F. Molisch, and J. Zhang, "Progressive accumulative routing in wireless networks," in *Proc. IEEE Global Communications Conference (GLOBECOM)*, Nov. 2006.

[11] T. Cover and J. A. Thomas, *Elements of Information Theory*, New York: Wiley, 1991.

[12] J. Gomez-Vilardebo, "Routing in accumulative multi-hop networks," in *2015 IEEE Conference on Computer Communications (INFOCOM)*, April 2015, pp. 1814–1821.

[13] ——, "Heuristic routing algorithms for minimum energy cooperative multi-hop wireless networks," in *European Wireless 2014; 20th European Wireless Conference; Proceedings of*, May 2014, pp. 1–5.

[14] M. Thakur and R. Tripathi, "Linear connectivity problems in directed hypergraphs," *Theoretical Computer Science*, vol. 410, pp. 2592 – 2618, 2009. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0304397509002011

[15] R. Urgaonkar and M. Neely, "Optimal routing with mutual information accumulation in wireless networks," *Selected Areas in Communications, IEEE Journal on*, vol. 30, no. 9, pp. 1730 –1737, october 2012.

[16] Y. H. Ezzeldin, M. Cardone, C. Fragouli, and D. Tuninetti, "Efficiently finding simple schedules in gaussian half-duplex relay line networks," *CoRR*, vol. abs/1701.04426, 2017. [Online]. Available: http://arxiv.org/abs/1701.04426

[17] A. Avestimehr, S. Diggavi, and D. Tse, "Wireless network information flow: A deterministic approach," *Information Theory, IEEE Transactions on*, vol. 57, no. 4, pp. 1872–1905, 2011.

[18] P. Razaghi and W. Yu, "Parity-forwarding for multiple-relay networks," in *Proc. IEEE Int. Symp. Information Theory (ISIT)*, July 2006, pp. 1678–1682.

Jesús Gómez-Vilardebó received his M.Sc. and Ph.D. degrees in Telecommunication Engineering from the Universitat Politècnica de Catalunya (UPC) in October 2003 and July 2009, respectively. In September 2005, he was granted by the Centre Tecnològic de Telecomunicacions de Catalunya (CTTC/CERCA) to obtain the Ph.D. on Signal Theory and Communications at the UPC. He is now with the CTTC holding a Research Associate position. His research interests are in the areas of information and communication theory, and their applications in multi-user communications, coded caching and information privacy.