

Advanced Approach for Directed Diffusion and
Network Coding

INE 3920 Thesis

Supervisor: Professor Victor, WEI

Student Name: Chow Ka Wai
Student ID: 01570243
Major: INE

Content

Abstract	P.3
Section 1	
Introduction	P.3
Section 2	
Overview of Directed Diffusion	P.4
1. Algorithm of Directed Diffusion	P.4
2. Differences between traditional networks	P.5
3. Advantages of Directed Diffusion	P.6
4. Limitations and disadvantages of Directed Diffusion	P.6
Section 3	
Overview of Network Coding	P.7
1. Algorithm of Linear Network Coding	P.7
2. Advantages of Network Coding	P.8
3. Limitations and disadvantages of Network Coding	P.9
Section 4	
Similarities and contrasts between Directed Diffusion and Network Coding	P.10
1. Data Propagation	P.10
2. Routing Layer	P.13
3. Local Interaction	P.14
4. Data Progression	P.15
Section 5	
Discussion	P.19
Section 6	
Conclusion	P.20
References	P.21

Abstract

Sensor networking becomes more and more important nowadays. Robust sensor networks can be deployed in many different usages, for example, in geographical use and in medical use. Therefore, a reliable and efficient paradigm is needed for the sensor networks. In this paper, we explore the Directed Diffusion paradigm for such coordination. Directed Diffusion is data-centric in that all communication is for named data. All nodes in a directed diffusion-based network are application-aware. This enables diffusion to achieve energy savings by selecting empirically good paths and by caching and processing data in-network. We also explore the Network Coding. The notion of network coding is to allow and encourage mixing of data at the intermediate nodes when information is multicast in a network. By comparing the Directed Diffusion and Network Coding, We try to combine these two paradigms to improve the throughput, bandwidth and energy issues that occur in the sensor network data transmission.

I. INTRODUCTION

Since sensor networks differ from traditional networks in several ways: sensor networks have severe energy constraints, redundant low-rate data, and many-to-one flows. The end-to-end routing schemes that have been proposed in the literature for mobile ad-hoc networks are not appropriate under these settings. Therefore, traditional routing algorithm cannot be used in sensor networks.

The Directed Diffusion is significantly different from the traditional address-based communication where nodes are identified by their destination addresses, and inter-node communication is layered on an end-to-end delivery service provided within the network. It uses in-network process (aggregation) to reduce the traffic in the sensor network. There has been proved that Direction Diffusion has noticeably better energy efficiency, especially in highly dynamic network. Network Coding introduces a new dimension into the information problem. Traditionally, only the routing dimension is considered in a transmission strategy includes both dimensions together is necessary to achieve the maximum information rate.

In this paper, we first describe the algorithm of Directed Diffusion, the differences with traditional networks, its advantages and disadvantages in section 2. Then describe the algorithm of Network Coding, together with its advantages and disadvantages in section 3. In section 4, we will compare the Directed Diffusion and Network Coding in four different fields. In section 5, we propose a new algorithm for sensor networks.

II. OVERVIEW OF DIRECTED DIFFUSION

Directed Diffusion is a new data dissemination paradigm for sensor network. It is significantly different from IP communication. The goal of Directed Diffusion is to establish efficient n-way communication between nodes. Directed Diffusion is a kind of low-level communication. It is data-centric and neighbor-to-neighbor. All data generated by sensor nodes using Directed Diffusion is named by attribute-value pairs. In addition to attribute-based naming, in-network processing for data aggregation and propagation (localized interactions) is also enabled in Directed Diffusion.

1. Algorithm of Directed Diffusion

A sink node requests data by sending interests for named data. Data matching the interest in the source node is then drawn down towards that sink node. Intermediate nodes can cache, or transform data, and may direct interests based on previously cached data.

Directed Diffusion consists of three phases:

Interest Diffusion:

When a sink node generates an interest, it will periodically broadcast the interest message to each of its neighbors. Since every node maintain an interest cache for suppressing duplicate messages and preventing looping, when intermediate nodes receive the broadcast interest, they will check to see if the interest exists in the interest cache. If there is no matching exists, the intermediate nodes will create an interest entry. Then they will re-broadcast the interest to their neighbors inside the network. Moreover, the intermediate nodes will setup a gradient towards the neighbor from which the interest received, a gradient represents both the direction towards which data matching an interest flows, and the status of that demand.

Exploratory reply:

All nodes received the interest will search the result until it meets the interest. If a node (source node) finds the result, it will send out the exploratory to neighbor nodes. The exploratory, containing data, flow downstream along the gradients set up by the interest. Also, there is a data cache inside every sensor node for data caching. It functions like the interest cache.

Reinforcement path:

Since the sink has multiple neighbors, it chooses to receive the data message for the same interest from a preferred neighbor. To do this, a reinforcement path is used, the sink reinforces the preferred neighbor nodes (for example, the node which delivered the data message with lowest latency), which, in turn reinforces its preferred

upstream neighbors, and so on. After a reinforcement path is established between the sink and the source, the source data will be delivered from the source to the sink through the reinforcement path. There is always more than one path being reinforced for one interest in sensor network. To negatively reinforce other paths, a mechanism negative reinforcement is used. With this approach, the sink can negatively reinforce its current preferred neighbor if another neighbor delivers better (lower latency) sensor data. Moreover, the sensor node can use this mechanism to locally repair the failed path. This negative reinforcement propagates neighbor-to-neighbor, removing gradients and tearing down an existing path if it is no longer needed. Negative reinforcements suppress loops and duplicate paths that may arise due to network dynamic.

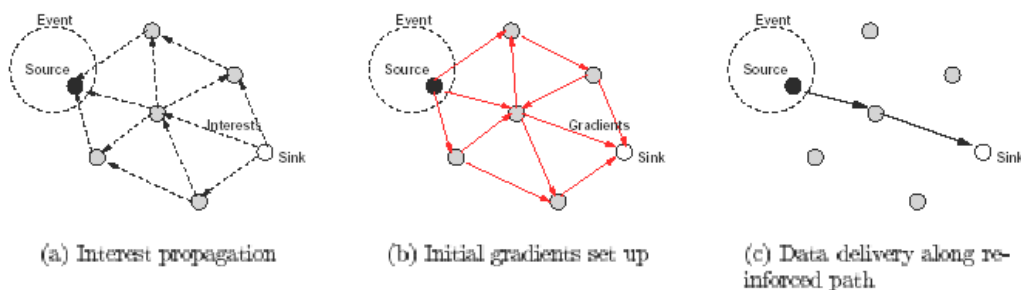


Fig. 1 A simplified schematic for Directed Diffusion [1]

2. Differences between traditional network

Directed Diffusion is clearly related to traditional network data routing algorithms, but there are several key features which differ from traditional networking. First, Directed Diffusion is data-centric, all communication in a diffusion-based sensor network uses interest to specify named data. Second, all communication in diffusion is neighbor-to-neighbor or hop-by-hop, unlike the traditional data networks with end-to-end communication. In other words, every node is an “end” in a sensor network. Third, there are no routers in a sensor network. Each sensor network can interpret data and interest message. This design choice is justified by the task-specificity of sensor networks. Sensor networks are not general purpose communication network. Fourth, sensor nodes do not need to have globally unique address. Nodes, however, do need to distinguish between neighbors. Finally, in an IP-based sensor network, for example, sensor data collection and processing might be performed by a collection of specialized servers which may, in general, be far removed from the sensed phenomena. In our sensor network, because every node can cache, aggregate, and more generally, process message, it is possible to perform

coordinated sensing close to the sensed phenomena.

3. Advantages of Directed Diffusion

There has been proved that Direction Diffusion has noticeably better energy efficiency, especially in highly dynamic network. It is because the data is transmitted from neighbor to neighbor, no data is propagated across the network. For some sensor fields, its dissipated energy is only 60% that of omniscient multicast. Moreover, every delivery has less than 20% additional average delay. [1] Furthermore, the application specific data aggregation in Directed Diffusion shows the benefit of in-network processing. An experiment comparing traffic with and without suppression has been proved that suppression is able to reduce traffic. Therefore, it can reduce the bandwidth needed for sensor networks. [2] Also Directed Diffusion is a robust dissemination in dynamic sensor networks, while at the same time minimizing the per-node configuration that is characteristic of today's network.

4. Limitations and Disadvantages of Directed Diffusion

There is limit memory storage for data caching inside the sensor node. Therefore, data aggregation maybe affected. Run time costs of matching. The cost of attribute matching is linear with the number of elements. The cost of matching as the number of attributes in one attribute set increase in different way [2].

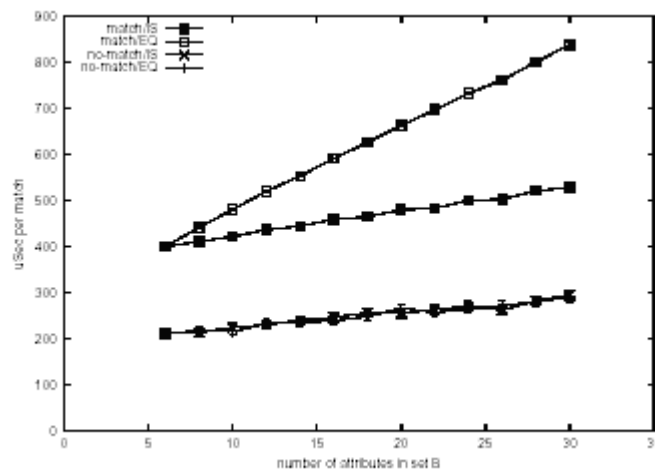


Fig.2 Matching performance as the number of attributes grow [2]

III. OVERVIEW OF NETWORK CODING

The throughput of information transmission within a data network is constrained by the network topology and link capacities. Therefore, to increase the throughput of information transmission, not only routing strategies, but also for coding strategies are needed. It is necessary to consider encoding/decoding data on nodes in the network, in order to achieve the optimal throughput. Since these coding operation are not restricted to source or destination nodes, they are referred to as network coding [19]. The notion of network coding is to allow and encourage mixing of data at the intermediate nodes when information is multicast in a network. Consider a communication network in which certain source nodes multicast information to other nodes on the network, a lot of data will be transmitted through the network. In order to reduce the traffic in the network and decrease the transmission delay, Network Coding is used. It allows a node to encode its received data before passing it on. After coding, the data size can be reduced.

In this paper, we will discuss the Linear Network Coding, the simplest coding scheme, which regards a block of data as a vector over certain base field and allows a node to apply a linear transformation to a vector before passing it on. It is sufficient to implement any feasible multicast connection. And routing and switching can be viewed as special cases of coding.

1. Algorithm of Linear Network Coding

The principle of Network Coding is easiest explained with the example. In the example, two sources having access to bits A & B at a rate of one bit per unit time have to communicate these bits to two sinks so that both sinks receive both per unit time. All links have a capacity of one bit per unit time [18].

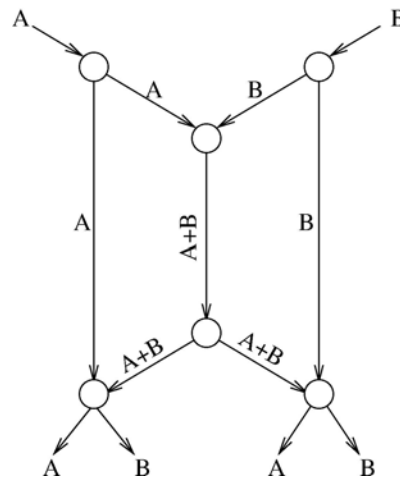


Fig.3 Network Coding [18]

However, linear network coding not only forwarding of bits at intermediate packet nodes, it is used to encode the data before transmitted. Assume in a two communication network, we want to multicast two bits b_1 and b_2 from the source S to both the nodes Y and Z . A solution is to let the channels ST, TW, TY carry the bit b_1 , channels SU, UW, UZ carry the bit b_2 , and channels WX, XY, XZ carry the exclusive-OR $b_1 \oplus b_2$. Then, the node Y receives b_1 and $b_1 \oplus b_2$, from which the b_2 can be decoded. Similarly, the node Z can decode the bit b_1 from b_2 and $b_1 \oplus b_2$. The coding/decoding scheme is assumed to have been agreed upon beforehand. [16]

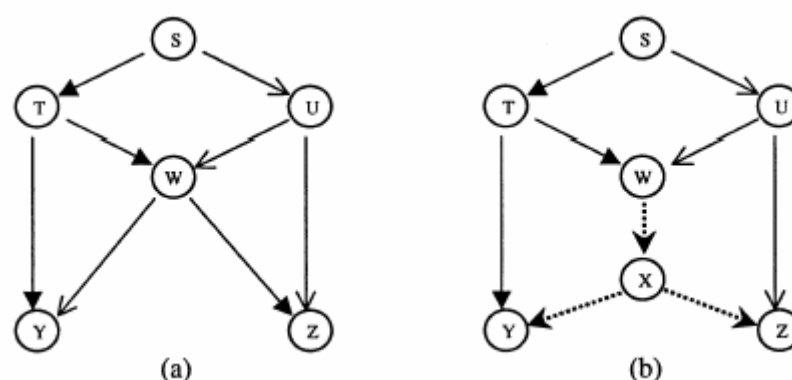


Fig. 4 Two communication networks [16]

2. Advantages of Network Coding

Network coding can be used to save bandwidth. In Fig. 5b, a total of 9 bits are sent. If network coding is not allowed, at least one more bit has to be sent in order that for t_1, t_2 and t_3 to recover both b_1 and b_2 . Thus, in this simple example using network coding can save 10% bandwidth [15]. Network coding can be used to improve throughput. For a multicast transmission in an undirected network, throughput can be improved by using network coding without random intermittent errors. However, the coding advantage, *i.e.*, the ratio of throughput improvement due to network coding, is always bounded by a constant factor of 2 in undirected network [17], [19]. Network coding can achieve the optimal throughput more easily. Including in both directed network and undirected networks, with both integral and fractional routing, optimal throughput with network coding is much more amenable to compute than optimal throughput without coding [19]. Network coding can be used to reduce the computational complexity for computing and achieved the optimal throughput. It

has been showed in information exchange, the computational complexity for computing are reduced. In multicast with integrated routing and multicast with fractional routing, the optimal throughput can be achieved [19].

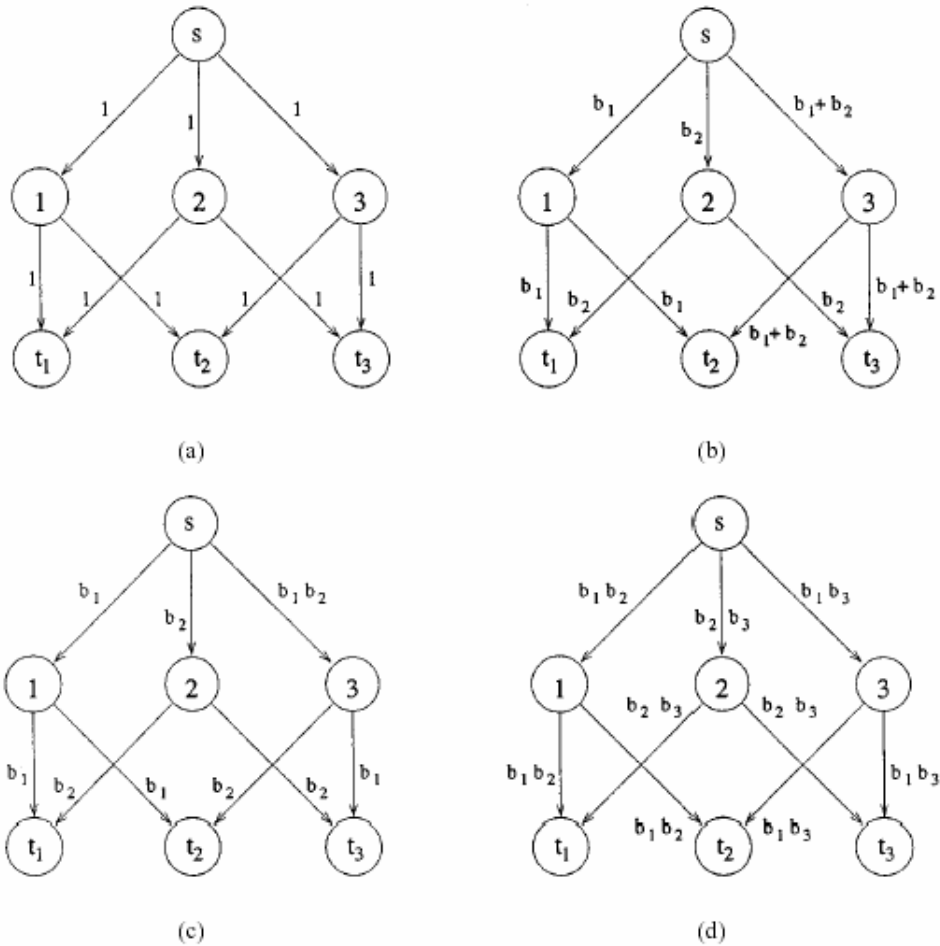


Fig. 5 A one source three sink network [15]

3. Limitations and Disadvantages of Network Coding

Network coding will increase the end-to-end latencies because of seeking occur in every intermediate node [13]. Network coding cannot increase the throughput. For unicast and broadcast networks in an undirected network, the coding advantage is always 1. There is no potential for network coding to improve the throughput. Moreover, for multicast network, the coding advantage is usually much smaller than the theoretical bound 2 [19].

IV. SIMILARITIES AND CONTRASTS BETWEEN DIRECTED DIFFUSION AND NETWORK CODING

In section 2 & 3, we have given the brief introduction about Directed Diffusion and Network Coding and shown their advantages and limitations. In this section, we will introduce some similarities and contrasts between Directed Diffusion and Network Coding, namely, Data Propagation, Routing Layer, Local Interaction and Data Processing.

1. Data Propagation

Directed Diffusion and Network Coding have totally different data propagation method. Directed Diffusion will broadcast the interests and use gradient and reinforcement for matching data transfer. Network Coding has multicast transmission in order to increase the throughput.

Directed Diffusion

Interest and Data Propagation:

Directed Diffusion can be divided into interest propagation and data propagation. In the interest propagation, the interests will be broadcasted by the sink node periodically to each of its neighbors. Since every nodes maintain an interest cache. The cache stores the interests received for aggregation and loop prevention. The gradients are established in the case where interests are flooded through a sensor network. The gradient specifies both a data rate and a direction in which to send events. For the intermediate nodes, they re-broadcast the interests to other neighbors. In data propagation, once sources detect a matching target, they send low-rate events, possibly along multiple paths, towards the sink. After the sink starts receiving these low data rate events, it reinforces one particular neighbor to draw down real data by selecting an empirically low delay path. It is very reactive to changes in path quality, whenever one path delivers an event faster than others, the sink attempts to use this path to draw down high quality data. However, because it is triggered by receiving one new event, this would be wasteful of resources. More sophisticated local rules are also possible. The algorithm described above can result in more than one path being reinforced in data propagation. For example, if the sink reinforces neighbor A, but then receives a new event from neighbor B, it will reinforce the path through B. If the path through B is consistently better, we will use negative reinforcement to negatively reinforce the path through A.

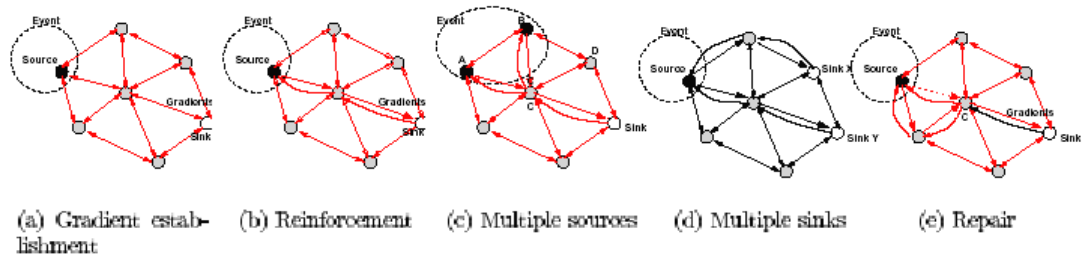


Fig.6 Illustrating different aspects of diffusion [1]

Negative Reinforcement:

Negative Reinforcement is to time out all high data rate gradients in the network unless they are explicitly reinforced and remove looping. Although the looping message will be immediately suppressed using a message cache, in general, we would still benefit from truncating the looping paths for resource savings. However, such loop removal is not always appropriate, specifically for some shared high-rate gradient maps with multiple sources and sinks. For example (Fig. 7(c)), if both sources send distinguishable events, the gradient **B-C** and **C-B** should not be truncated because each of them is necessary for delivering events for a particular source-sink pair. Although such gradients may deliver some looping events, they also consistently deliver new events. With our conservative rule for negative reinforcement, those gradients will not be negatively reinforced [21].

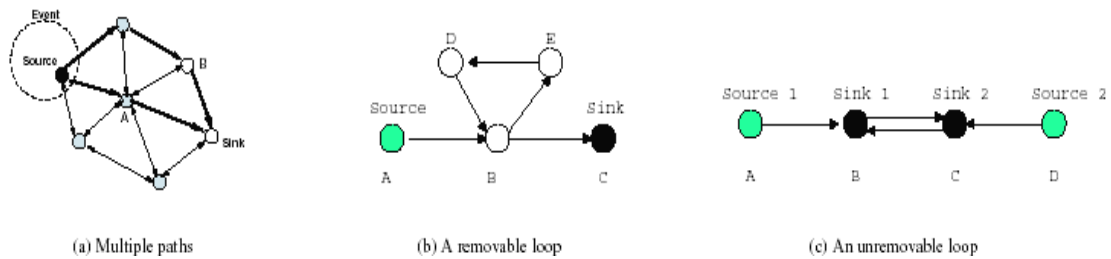


Fig.7 Negative reinforcement for path truncation and loop removal [21]

Local Repair for Failed Paths:

In Directed Diffusion, intermediate nodes on a previously reinforced path can apply the reinforcement rules. This is useful to enable *local repair* of failed or degraded paths. Causes for failure or degradation include node energy depletion, and environmental factors affecting communication (*e.g.*, obstacles). Consider Fig. 6(e), in which the quality of the link between the source and node **C** degrades and events are frequently corrupted. When **C** detects this degradation—either by noticing that the event reporting rate from its upstream neighbor (the source) is now lower, or by

realizing that other neighbors have been transmitting previously unseen location estimates—it can apply the reinforcement rules to discover the path shown in the figure. Eventually, C negatively reinforces the direct link to the source (not shown in the figure). Our description so far has glossed over the fact that a straightforward application of reinforcement rules will cause all nodes downstream of the lossy link to also initiate reinforcement procedures. This will eventually lead to the discovery of one empirically good path, but may result in wasted resources. One way to avoid this is for C to interpolate location estimates from the events that it receives so that downstream nodes still perceive high quality tracking [21].

Network Coding

In Network coding, each application layer node communicates only with its neighbors and transfer data along the edges in the overlay network. Multicasting is implemented by forwarding messages along overlay multicast trees that are constructed and embedded in the virtual overlay network. In [13], a multicast graph is proposed for Network Coding. During the construction process, data delivery paths should be optimized in the multicast graph to the receivers. By using a multicast graph instead of a multicast tree, it can maximize multicast performance (end-to-end throughput and latency) and minimize the penalty incurred by elevating the functionality of multicast from the IP layer to the application layer. Since using a multicast graph or a multicast tree, no looping will occur.

Recovery from network failures:

Coding is not only applicable to networks in order to achieve capacity, but can also be used to recover from network failures. Such failures are different from random errors, such as packet losses or bit errors on links, which are described by probabilistic processes. The failures we consider entail the permanent removal of an edge, such as would occur in a network if there were a long-term failure due to a link cut or other disconnection. Currently, such failures are dealt with through the use of rerouting, such as link or path protection. Coding can also be used to protect against link failures in networks [17]

2. Routing Layer

In order to reduce the transmission of data information and increase the throughput in data or sensor network, the Direction Diffusion paradigm and Network Coding both are employed in application layer.

Direction Diffusion

Since sensor networks differ from traditional network in several ways: sensor network have severe energy constraints, redundant low-rate data, and many-to-one flows. Therefore, the end-to-end network layer routing schemes that have been proposed in the literature for mobile ad-hoc networks are not appropriate under these settings.

Ad hoc routing recreates IP style addressing. It adds substantial overhead when applying this routing to highly resource-constrained environments such as sensor networks. For example, some approaches to service location for smart places require service for IP assignment, IP-level routing, host name lookup, and service registration and lookup. And ad hoc routing has end-to-end process only, it does not support in-network processing.

Directed Diffusion also cannot employ in multiple layers. Multiple layers of naming may not be a bottleneck with a few or even tens of nodes, but the overhead becomes unreasonable with hundreds or thousands of nodes that vary in availability (due to movement and failures). However, sensor networks can profit by eliminating multiple layers and naming by routing data directly in application terms. Efficient attribute naming is based on external frames of reference such as predefined attributes and geography. Predefined sensor types reduce the levels of run time binding and geographic-aided routing reduces resources consumption. Moreover, in-network processing is also supported [2].

Network Coding

By applying the concept of network coding in application multicast, motivating the case for application-layer coded multicast. The objective is to taking advantage of alternate paths and excess capacity in the IP-layer network topology, and to significantly increase end-to-end multicast capacity.

Each application-layer node communicates not only with its neighbors in the overlay network. Multicasting is implemented by forwarding messages along multicast trees that are constructed and embedded in the virtual overlay network. Application –layer multicast, in general, enjoys two attractive advantages over traditional IP multicast: 1. Multicast support in the network layer is not required. 2.

Data is transmitted between nodes via unicast, effectively exploiting all existing security, flow control and reliable delivery mechanisms that are readily available and mature. However, an overlay multicast approach cannot perform as efficient as IP multicast. It is impossible to completely prevent multiple overlay edges from traversing the same physical link, causing unavoidable redundant traffic on the same link. Further, unicast communication between end systems involves traversing other end systems, potentially increasing latency. It is therefore critical to evaluate and seek to minimize both the relative increase of end-to-end latencies and the increase in per-link bandwidth requirements as compared with network-layer multicast.

In short, both Directed Diffusion and Network Coding can get large advantages by applying them in application layer.

3. Local Interaction

Local interaction means there are messages exchange between neighbors or nodes within some vicinity. For Directed Diffusion, the interests and data communication is neighbor-to-neighbor (node-to-node), unlike the end-to-end communication in traditional data networks. For Network Coding, the data communication is end-to-end, and there is no local interaction.

Directed Diffusion

Localized interaction is an important feature of Directed Diffusion. Since energy efficiency is one of the most important requirements for sensor network, the short range node-to-node communication is preferred over long range communication to the destination. This can be energy efficient in highly dynamic networks when change in topology need not be propagated across the network. By performing local computation to reduce data before transmission can also obtain orders of magnitude energy savings.

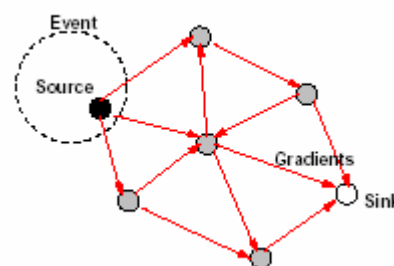


Fig. 8 Initial gradient setup [1]

The interests and data propagation and aggregation are determined by localized interactions. Every pair of neighboring nodes in sensor network establishes a gradient towards each other. After receiving an interest and data, a node has no way of knowing whether that interest was in response to one it sent, and it may decide to resend the interest and data to some subset of its neighbors. To its neighbors, this interest appears to originate from the sending node, although it might be come from a distant sink. This is an example of a local interaction. In this manner, interests diffuse throughout the network. Not all received interests are resent. A node may suppress a received interest if it recently resent a matching interest.

Network Coding

With Network Coding, nodes have the capability of encoding and decoding data at the per-message level using efficient linear codes. It is a type of end-to-end communication. It is similar to the IP-layer network. When the sender sends out the data to multiple receivers, in order to perform coding, alternate paths instead of the shortest widest paths in the multicast tree are used. The links between nodes can be dynamically created or torn down to construct topologies that are conducive to better network performance. That means not every nodes in the network will receive the data. Therefore, there is no local interaction in Network Coding.

4. Data Processing

There are always many data transferred by the sensor nodes in the sensor network. Moreover, these data are distributed across the entire sensor network, and so are hard to use. Communication between the nodes in the sensor network requires the expenditure of energy, a scarce commodity in most sensor networks. Thus, making effective use of sensor network data will require scalable, self-organizing, and energy efficient data dissemination algorithms. Directed Diffusion and Network Coding also have data processing before transmit data to neighbor nodes. However, their data process methods are different. Directed Diffusion uses aggregation as its main data processing method. Network Coding not only uses aggregation but also encoding/decoding as its data processing methods.

Directed Diffusion

Since the content of data is more important than the identity of the nodes in the

sensor network, Directed Diffusion paradigm shifts the focus from the traditional address-centric approaches (finding the shortest routes between pairs of addressable end-nodes) to a more data-centric approach (finding routes from multiple sources to a single destination that allows in-network consolidation of redundant data). That means routing decisions are based on the name of the data rather than on the identity of sending and receiving nodes [6], [20].

Directed Diffusion can be divided into two main phase, namely the interest propagation phase, where interest message flow from the sink to the source and the data propagation phase where data messages flow from the source to the sink. Data sources and sinks use attributes to identify what information they provide or are interested in. In interest propagation, an interest is a list of attribute-value pairs that describe a task using some task-specific naming scheme. The attributes describe the data that is desired by sensor types and possibly some geographic region. They are then used to identify and contact all relevant sensors. Then the interests will be flooded in the network. When a sensor node that matches the interest is found, the application activates its local sensors to begin collecting data. The sensor node then generate data message matching the interest. And it becomes data propagation phase. The data message is also represented using an attribute-based naming scheme. Since there maybe some sources send data to the sink at the same time, the intermediate nodes will store the data received in its own cache before propagating toward sink. When the data is cached, the intermediate nodes will perform some form of aggregation/consolidation function on the data origination at multiple sources. The core diffusion mechanism uses the cache to suppress the duplicate messages, prevent loops, perform data processing and it can be used to be preferentially forward interests. For example, data from detections of a single object by different sensors may be merged to a single response based on sensor specific criteria [2].

In Fig 9, if sources 1 and 2 both send the same data, data aggregation will occur at node B. Node B will send only one of these forward.

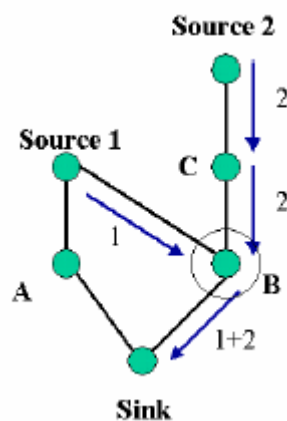


Fig. 9 Intermediate node B can perform data aggregation [20]

Performance measures by using in-network data processing:

- **Energy Savings:** By aggregating the information coming from the sources, the number of transmissions is reduced. Since traffic represents both data and control traffic. Comparing traffic with and without suppression shows that suppression is able to reduce traffic by up to 42% for four sources [2]. Therefore, there is a savings in energy and bandwidth.
- **Delay:** The potential disadvantage of data processing is increased delay. The delay highly depends on the aggregation function, for some simple kind of data aggregation such as duplicate suppression. There is no need for data to be withheld at an aggregating node. For more complicated forms of data aggregation, where the output aggregated packet depends on the combination of multiple input packets this delay is an issue [20], [2].
- **Robustness:** Because the data processing reduces the energy cost of data transmission. This can be considered as providing some degree of robustness to dynamics in the sensed phenomena.

Network Coding

In existing computer networks, each node functions as a switch in the sense that it either relays information from an input link to an output link, or it replicates information received from an input link and sends it to a certain set of output links. Furthermore, a node can function as an encoder in the sense that it receives information from all the output links. By employing network coding at the nodes, bandwidth can in general be saved.

Data transferred in Network Coding is not data-centric. Apart from aggregation, encoding/decoding is also be used in data processing. There is a theorem used in Network Coding, Max-flow Min-cut Theorem, which characterizes the admissible coding rate region for the single-source problem. We illustrate the Network Coding by two examples. Fig. 10(a) shows the capacity of each. It is easy to check that the value of a max-flow from s to t_1 is 2, so by using Max-flow Min-cut theorem, we can send 2 bits b_1, b_2 to t_1, t_2 simultaneously. And Fig. 10(b) shows such a scheme, where “+” denotes the modulo 2 addition. At t_1 , b_2 can be recovered from b_1 and b_1+b_2 . Similarly, b_1 can be recovered at t_2 . Note that when there is more than one sink, we can no longer think of information as a real entity, because information needs to be replicated or transformed at the nodes. In this example, information is coded at node 3, which is unavoidable [15].

In Fig. 5, we investigate the savings in bandwidth when Network Coding is

allowed. In Fig. 5(b), a total of 9 bits are sent. If Network Coding is not allowed, then it is easy to see that at least one more bit has to be sent in order that for t_1, t_2, t_3 to recover both b_1, b_2 . Thus we see that a very simple network code can save 10% in bandwidth [15]. Moreover, Network coding can increase the throughput under some limitations and achieve the optimal throughput more easily [19].

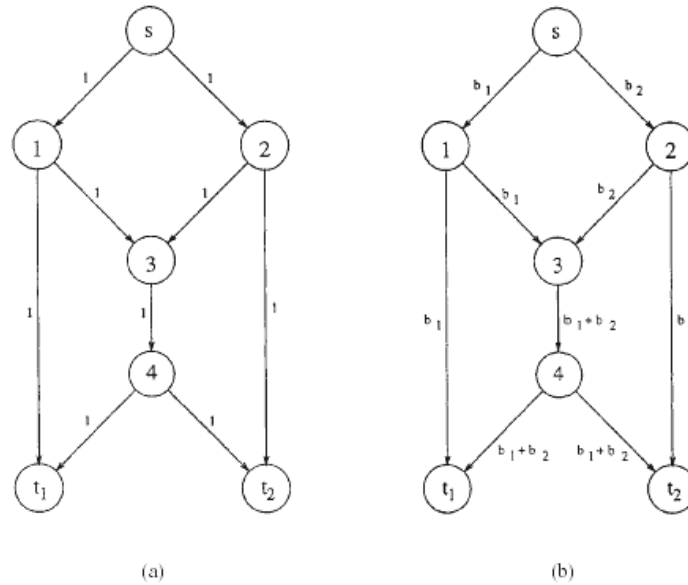


Fig. 10 A one source two sink network with coding [15]

In short, the data generated in Directed Diffusion is data-centric, but Network Coding is not. Directed Diffusion uses aggregation as its data progression method. Network Coding uses aggregation and encoding/decoding as its progress method. Both can save bandwidth and energy. And Network Coding can further achieve the optimal throughput.

V. DISCUSSION

After comparing the similarities and contrasts of Directed Diffusion and Network Coding, we try to introduce a new algorithm that combines the advantages of Directed Diffusion and Network Coding. The new algorithm should be run on the application layer. It is because the data generated by sensor nodes can be named by attribute-value pairs (data-centric). Therefore, the nodes can perform in-network processing. Since the sensor network is undirected network, where each communication link is bidirectional, using multicast in application layer can take advantage of alternate paths and excess capacity in the IP-layer network topology and to improve the throughput.

For interest propagation, our algorithm just likes the Directed Diffusion. The sink broadcast interests to every neighbor in the network. Although more bandwidth is needed for flooding the interests, it is the easiest way to send the interests to every node in the network. After the intermediate nodes receive the interest, every pair of neighboring nodes establishes a gradient towards each other. We use the concept from directed diffusion for gradient setup such that it can forward data along different paths and selection of the optimal path by considering not only the shortest path but also the high quality data paths. Moreover, setting up gradients can perform local interaction between nodes.

For data propagation from sources to sinks, our algorithm likes the Network Coding. The interests generated by the node carry the identity of the node. Therefore, the sources will know the interest was in response to one it sent out earlier. When the source detects the matching target, the source sends the data to the sink via multicast. And by using the gradients set up in interest propagation, a high quality data path is chose as data propagation path. The intermediate nodes have interest and data caches for aggregation and encoding/decoding. By sending the data via multicast, less bandwidth is needed and the optimal throughput can be achieved more easily.

However, there are some disadvantages. The end-to-end data transmission latency increase because of data progressing occurs in the intermediate nodes. Moreover, more energy may be used for data processing and flooding interests.

VI. CONCLUSION

In this paper, we described the Directed Diffusion algorithm and Network Coding. And we pointed out the advantages and disadvantages of Directed Diffusion and Network Coding. After that, we compared the features of Directed Diffusion and Network Coding in 4 different fields. Finally, we introduced a new algorithm by combination the Directed Diffusion and Network Coding.

Directed Diffusion has some novel features – data-centric dissemination, flooding, gradient establishment, reinforcement-based adaptation to the empirically best path, and in-network data aggregation and caching. Directed Diffusion can enable highly energy-efficient and robust dissemination in dynamic sensor networks, while at same time minimizing the per-node configuration [1].

Network Coding is an end-to-end communication. It allows a node to aggregate, encode and decode its received data before passing it on. The aim is to use bandwidth more efficiently and thereby increase network capacity.

For the algorithm we proposed, it includes some features of Directed Diffusion and Network Coding. For example, flooding and gradient establishment inherited from Directed Diffusion, encoding/decoding inherited from Network Coding. We believe that this combination of Directed Diffusion and Network Coding could save up a lot of bandwidth and energy and hence increase the throughput of the whole system.

References

- [1] Deborah Estrin , Ramesh Govindan and Chalermek Intanagonwiwat,
“Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks”
- [2] Deborah Estrin, Ramesh Govindan, Deepak Ganesan, John Heidemann,
Chalermek Intanagonwiwat and Fabio Silva, “Building Efficient Wireless Sensor Networks with Low-Level Naming”
- [3] Miodrag, Sasha Slijepcevic, Mani B. Srivastava, Vlasios Tsiatsis and Scott Zimbeck, “On Communication Security in Wireless Ad-Hoc Sensor Networks”,
(Proceedings of the Eleventh IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE’02)),
1080-1383/02, 2002
- [4] Gang Qu and Lin Yuan, “Design Space Exploration for Energy-Efficient Secure Sensor Network”, (Proceedings of the IEEE International Conference on Application-Specific Systems, Architectures, and Processors (ASAP’02)),
1063-6862/02, 2002
- [5] Luigi V.Mancini, Paul Havinga, Roberto Di Pietro, Sandro Etalle and Yee Wei Law, “LKHW: A Directed Diffusion-Based Secure Multicast Scheme for Wireless Sensor Networks”
- [6] Brad Karp, Deborah Estrin, Ramesh Govindan, Scott Shenker and Sylvia Ratnasamy, “Data-Centric Storage in Sensornets”
- [7] Ahmed E.Kamal, Jamal N.Al-Karaki and Raza UI-Mustafa, “Data Aggregation in Wireless Sensor Networks – Exact and Approximate Algorithms”
- [8] Paul Havinga and Supriyo Chatterjea, “CLUDDA – Clustered Diffusion with Dynamic Data Aggregation”
- [9] Bharat Bhargava, Leszek Lilien, Maleq Khan, Pankaj and Sarika Agarwal,
“Self-configuring Node Clusters, Data Aggregation, and Security in Microsensor Networks”
- [10] Bhasker Krishnamachari, Deborah Estrin and Stephen Wicker, “The Impact of Data Aggregation in Wireless Sensor Networks”
- [11] D.Muthuavinashiappan, H.Cam, P.Nair and S.Ozdemir, “ESPDA:
Energy-Efficient and Secure Pattern-based Data Aggregation for Wireless Sensor Networks”
- [12] Satya N.Majumdar, “Travelling Front Solutions to Directed Diffusion Limited Aggregation, Digital Search Trees and the Lempel-Ziv Data Compression Algorithm”
- [13] Baochun Li, Member, IEEE, Jisng Guo and Ying Zhu, “Multicast with Network

Coding in Application-Layer Overlay Networks”

- [14] David Karger, Michelle Effros, Muriel Medard and Tracey Ho, “On Coding for Non-Multicast Networks”
- [15] Rudolf Ahlswede, Ning Cai, Shuo-Yen Robert Li and Raymond W. Yeung, “Network Information Flow”, 0018-9948, 2000 IEEE
- [16] Ning Cai, Shuo-Yen Robert Li and Raymond W. Yeung, “Linear Network Coding”
- [17] Muriel Medard and Ralf Koetter, “Beyond Routing: An Algebraic Approach to Network Coding”
- [18] Ralf Koetter (koetter@uiuc.edu), “The Network Coding Home Page”, 10/29/2003, <http://www.comm.csl.uiuc.edu/~koetter/NWC/>
- [19] Baochun Li and Zongpeng Li, “Network Coding in Undirected Networks”
- [20] Bhaskar Krishnamachari, Deborah Estrin and Stephen Wicker, “Modelling Data-Centric Routing in Wireless Sensor Networks”
- [21] Chalermek Intanagonwiwat, Deborah Estrin, Fabio Silva, John Heidmann and Ramesh Govindan, “Directed Diffusion for Wireless Sensor Networking”