Weekly Report (2010-05-23)

Hongxing Li

I. Multi-hop Physical Network Coding in Wireless Networks

A. Motivation

Traditional digital network coding schemes are all successfully executed based on the prerequisite that each decoding node has $n-1$ original data packets, which are retrieved by overhearing or generated by itself, so as to decode the coded packet of n original data packets. Even with the physical network coding proposed in [1], the prerequisite still holds as the sufficient-and-necessary condition for successful decoding.

Meanwhile, all current network coding schemes are applied either on a butterfly-like structure (Fig. 1) for overhearing or a two-way transmission structure (Fig. 2) for self-generated original data packet. These two structure requirements put more restriction on exploiting the power of network coding. First, in butterfly-like structure with overhearing, destination node $D1$ ($D2$) should be in the transmission range of source node $S1$ ($S2$). If that requirement is not satisfied, where data packet from $S1$ ($S2$) needs multi-hop transmission to reach $D1$ ($D2$), overhearing cannot be executed which leads to no network coding benefit for this case. Second, in two-way transmission structure, node $S1$ and $S2$ should be transmitting on the opposite direction with node R as the relay. So network coding cannot enhance the performance when source nodes are transmitting along the same direction.

As a conclusion, we have no proper network coding scheme on hand to tackle the network throughput enhancement for multi-hop data flows not on the opposite direction while without overhearing opportunity.

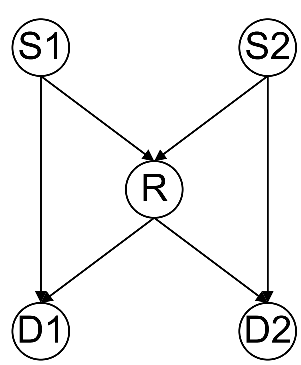


Fig. 1: Butterfly-like Structure.

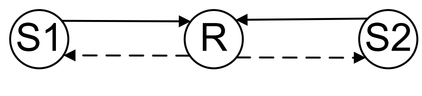


Fig. 2: Two-way transmission Structure.

B. *One-Shot Scenario*

In this report, I propose a novel network coding scheme working on the physical layer, which targets on the above issue. I will explain the details with an example in Fig. 5 and 4.

Here, we have two source nodes a and b with the same destination as node s which is out the transmission range of both a and b. Node c and d will act as the relay. In the example, we only consider one-shot transmission, which means each source only has one data packet for the destination. I will show the case for infinite data packets in later examples.

Since destination s cannot overhear any data packet and both sources are transmitting along the same direction, current network coding schemes cannot be applied. Fig. 5 presents how data packets are delivered to the destination in 4 time slots with traditional scheduling without network coding.

However, we can accomplish the transmission in 3 time slots with our novel physical-layer network coding. In Fig. 4(a), node a and b are scheduled to transmit simultaneously. Suppose the modulated signal at a (b) is x_a (x_b). Then, the received signal y_c (y_d) at relay node c (d) is as follows.

$$y_c = h_{ac} \cdot x_a + h_{bc} \cdot x_b + n_c$$

$$y_d = h_{ad} \cdot x_a + h_{bd} \cdot x_b + n_d$$

Here, h_{ij} represents the channel gain from node i to node j, which characterize the channel state information like attenuation, fading and frequency drift etc. n_i is the ambient noise at node i.

In Fig. 4(b-c), relay node c and d use two time slots to forward the received signal to node s with *amplify-and-forward* scheme. So, node s will receive two signals as follows,

$$y_s^{(1)} = h_{cs} \cdot a_{cs} \cdot y_c + n_s$$

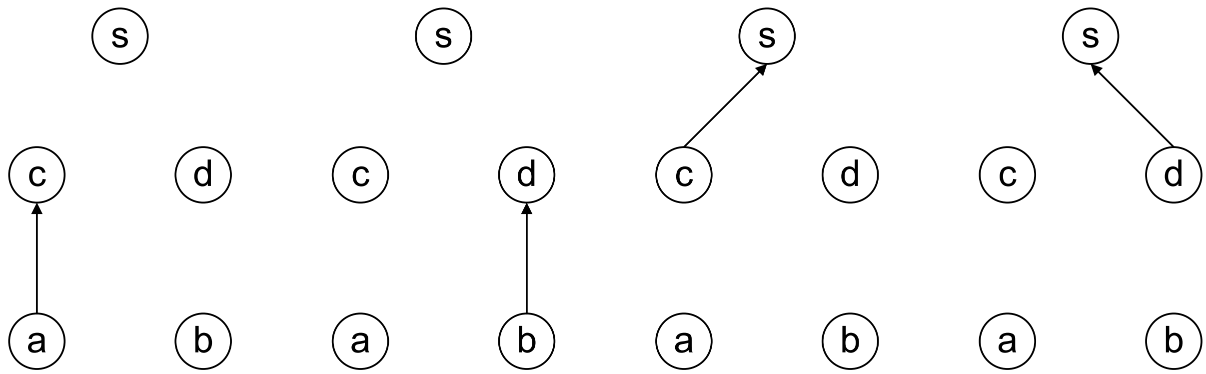
$$y_s^{(2)} = h_{ds} \cdot a_{ds} \cdot y_d + n_s$$

Here, a_{ij} is the multiplicative factor for signal amplification from i to j with *amplify-and-forward*.

So, at node s, we get two linear equations on variable x_a and x_b. If node s obtains the channel state and noise information in two hops, which is also assumed in [1] for physical network coding, we can solve the equations and decode two data packets.

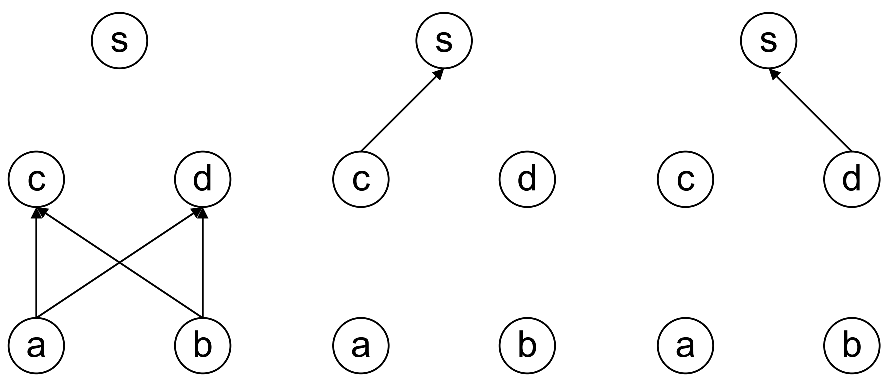
Note that, we can have multiple destinations for the above example. 3 time slots is still enough to complete the transmissions as long as all the destinations are in the transmission range of both relay node c and d, and be aware of the channel state and noise information in two hops.

On the other hand, two source nodes and two relay nodes can also decode the two data packets if they obtain the channel state and noise information since they also receive two combined signals like s does. So our work is also applicable for multiple-flow multicast or broadcast.



(a) Time slot 1 (b) Time slot 2 (c) Time slot 3 (d) Time slot 4

Fig. 3: 2-hop one-shot converge-cast with two sources and two relays under traditional scheduling.



(a) Time slot 1 (b) Time slot 2 (c) Time slot 3

Fig. 4: 2-hop one-shot converge-cast with two sources and two relays under C-network-coding.

C. Infinite Packet Scenario

Next, I extend our network coding scheme to multi-hop multi-source transmissions with infinite data packets as in Fig. 5–8. In this example, we have the left-most two nodes as the sources. They have infinite data packets to be delivered along the right-hand direction. We define one round of transmission to be the number of time slots between the source sending out the i^{th} and $(i+1)^{th}$ original data packets. For example, in fig. 5, the first packet is sent out at time slot 1 while the same source sends out the second packet at time slot 7. Then one round transmission without network coding is 6 time slots.

We classify the relay nodes into categories such that the relay nodes of at least i hop from the source are called i-hop relay nodes.

With physical network coding proposed in [1], we need 4 time slots for each round of transmission as shown in fig. 6. The dashed line in fig. 6(e) means that the signal can be decoded with physical network coding.

Here, I categorize our novel multi-hop physical network coding into two classes:

1) *global channel state information at the destination*: In this case, only the destination obtains all channel state information along the route. So the decoding process is only executed at the destination. In this way, we can skip the decoding process on relay nodes and carry out *amplify-and-forward* immediately. However, the drawback is that it is not realistic in wireless networks, especially ad-hoc networks, to get global information at one node. Besides, the relay node cannot recover the original data which means it is not applicable when any relay node is also in the set of destinations.
2) *local channel state information at each relay*: For this scenario, each relay maintains the two-hop local channel state information and conduct decoding. The execution detail is that $2i$-hop relay nodes receive combined signal from sources or $(2i-1)$-hop relay nodes and execute *amplify-and-forward* one by one to forward combined signal to $(2i+1)$-hop relay nodes or destinations. The $(2i+1)$-hop relay nodes will send out the recovered data packet concurrently to $(2i+2)$-hop relay nodes just in the same way as sources do. The benefit is that all relay nodes and source nodes can recover all original data packets.

I will show that, although global information is obtained, the case of global channel state information at the destination cannot outperform the one of local channel state information at each relay. In fig. 7, there are 3 time slots in one round of transmission for the case of global channel state information at the destination. Although relay nodes can conduct concurrent relaying in one time slot, e.g. fig. 7(b–d), each link along the same data flow should be separated by at least two hops to avoid interference since the relay nodes are not able to decode the combined signals.

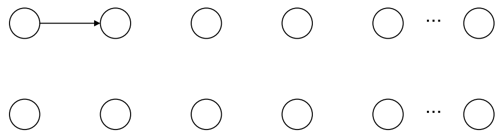
However, in fig. 8, we see that the number of time slots in one round of transmission for the case of local channel state information at each relay is also 3. Although the 1-hop relay nodes in fig. 8(b–c) should be scheduled one by one so that the next hop relay nodes can decode the data packets, the combined signal from the 2-hop relay nodes can be decoded at the 1-hop relay nodes. So each link along the same data flow should only be separated by one hop.

D. Comparison with similar works

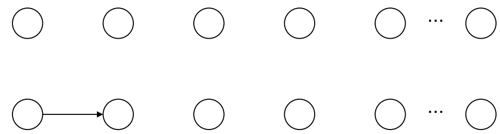
Complex-Field Network Coding:

In [2], a Complex-Field Network Coding scheme is proposed to conduct multi-source multi-relay single-destination scheduling similar to our examples. It is claimed that no matter how many sources are involved, all data packets can be delivered to the destination with 2 time slots.

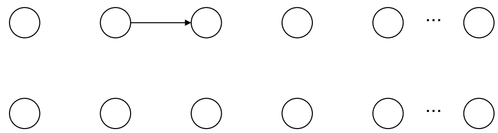
However, its conclusion is debatable for following reasons.



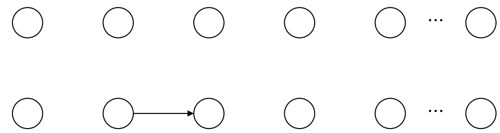
(a) Time slot 1



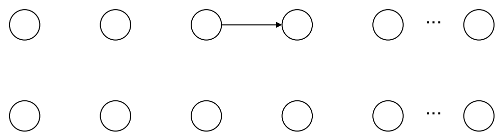
(b) Time slot 2



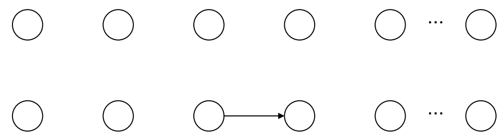
(c) Time slot 3



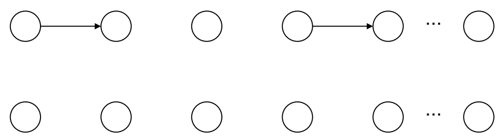
(d) Time slot 4



(e) Time slot 5

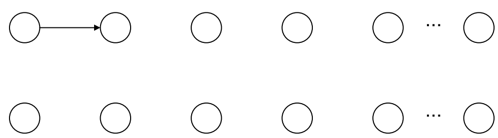


(f) Time slot 6

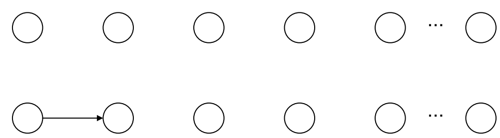


(g) Time slot 7

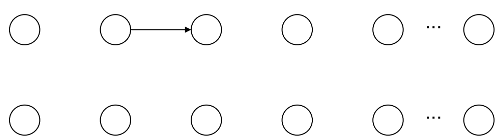
Fig. 5: Multi-hop continuous multicast without network coding.



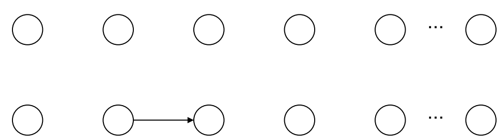
(a) Time slot 1



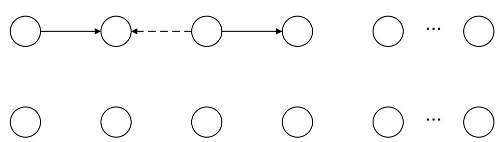
(b) Time slot 2



(c) Time slot 3



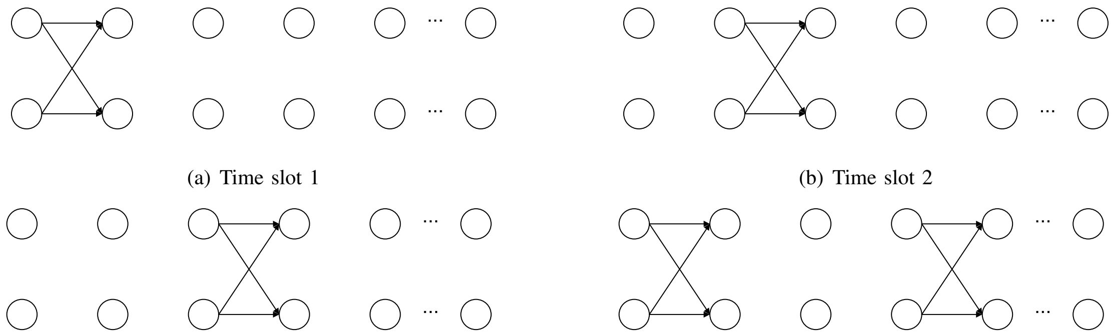
(d) Time slot 4



(e) Time slot 5

Fig. 6: Multi-hop continuous multicast with traditional physical network coding.

- One important assumption is that the destination node should be in the transmission range of all sources. So the targeted problem of [2] in essentially different from ours.



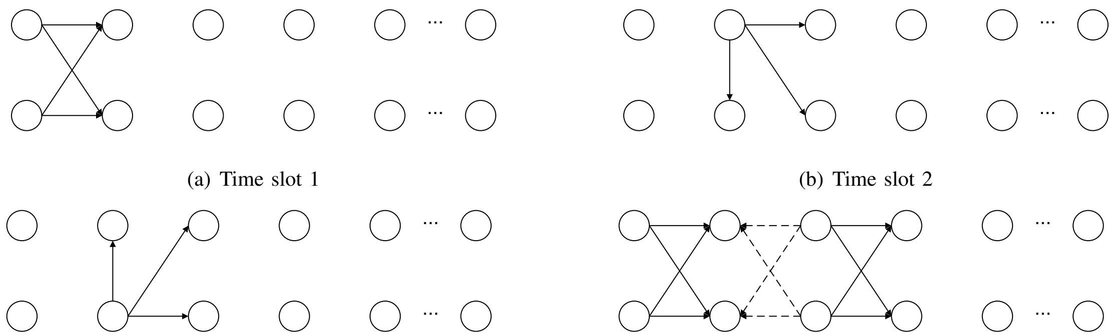
(a) Time slot 1

(b) Time slot 2

(c) Time slot 3

(d) Time slot 4

Fig. 7: Multi-hop continuous multicast with global channel state information at the destination.



(a) Time slot 1

(b) Time slot 2

(c) Time slot 3

(d) Time slot 4

Fig. 8: Multi-hop continuous multicast with local channel state information at the each relay.

- Its power stems from the *multi-user detection* technologies instead of network coding. In fact, the *maximum likelyhood* demodulation is applied as multi-user detection to estimate the original data packets. However, *maximum likelyhood* demodulation has a complexity of 2^n for n sources and generates great power penalty. Furthermore, the output of *maximum likelyhood* demodulation is just an estimation of original data with considerable estimation error.

So, we can clarify that [2]'s work is in a different field from ours.

General Algorithm for Interference Alignment and Cancellation:

But, there does exist one recent work [3] which is quite similar to our mechanism. In [3], a general algorithm for *interference alignment and cancellation (IAC)* is presented for multi-hop multi-source transmissions with relay nodes.

Fig. 9 servers as an illustration of how the *IAC* is applied in multi-hop wireless networks in [3]. S_1 and S_2 are two sources with d_1 and d_2 as the destination respectively. In time slot 1 and 2, S_1 and S_2 send data packet x_1 and x_2 respectively to u_1, u_2 and u_3. In time slot 3, u_1 and u_2 relay x_1 and x_2 respectively to d_1 and d_2 while u_3 relays an linear combination of x_1 and x_2 such that the portion of x_1 (x_2) in the received signal can be canceled at d_2 (d_1).

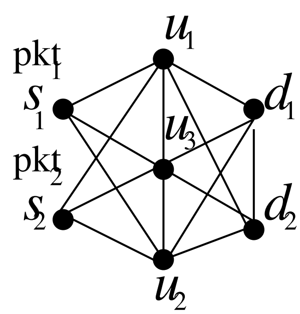


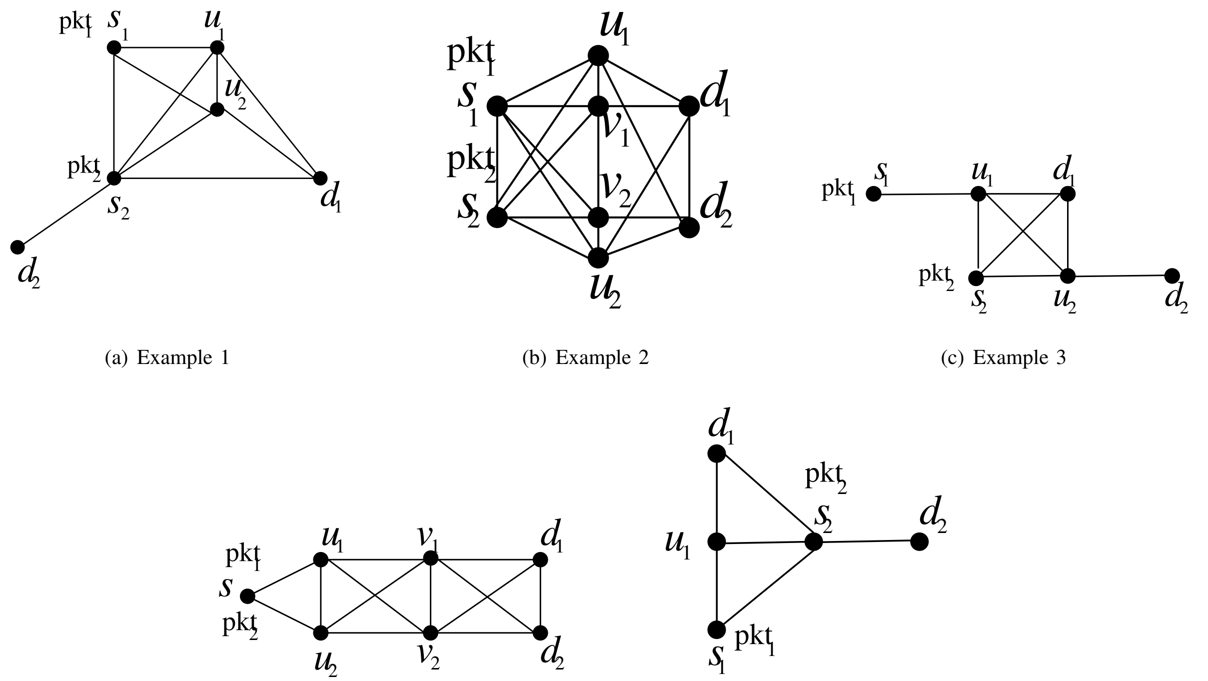
Fig. 9: Illustration for Interference Alignment and Cancellation.

In [3], five example (Fig. 10) are presented to show the power of *IAC* in improving the network throughput. I will compare our novel physical network coding with *IAC* in these examples as follows.

1) *Example 1*: *IAC* needs two time slots. In time slot 1, S_1 transmits to u_1 and u_2, and S_2 transmits to u_1, u_2, d_1 and d_2. In time slot 2, u_1 and u_2 transmit to d_1 with *IAC*. Our scheme also uses 2 time slots. However, we do not need two relay nodes. Any one out of u_1 and u_2 is enough since d_1 has overheard packet x_2 from S_2 and can recover S_1's packet x_1 with network coding.
2) *Example 2*: *IAC* needs two time slots. In time slot 1, S_1 and S_2 transmit. In time slot 2, u_1, u_2, v_1 and v_2 transmit. v_1 (v_2) will null x_2's (x_1's) signal from u_1 (u_2). Our scheme uses 3 time slots. In time slot 1, S_1 and S_2 are scheduled. In time slot 2 and 3, u_1 and u_2 are scheduled respectively. Although we need one more time slot, we only require two relay nodes, which reduces the energy consumption.
3) *Example 3*: Both *IAC* and our scheme need two time slots. In time slot 1, S_1 and S_2 are scheduled. And u_1 and u_2 are scheduled in time slot 2.
4) *Example 4*: *IAC* needs four time slots. In time slot 1 and 2, S is scheduled. In time slot 3, u_1 and u_2 are scheduled. And v_1 and v_2 are scheduled in time slot 4. Our scheme uses 5 time slots. In time slot 1 and 2, S is scheduled. u_1 and u_2 are scheduled in time slot 3. In time slot 4 and 5, v_1 and v_2 transmit respectively. Although our scheme needs one more time slot, v_1 and v_2 can also recover the data packets while it is not possible with *IAC*.
5) *Example 5*: Both *IAC* and our scheme need two time slots. In time slot 1, S_1 and S_2 are scheduled concurrently. And in time slot 2, u_1 transmits.

From the above case-study, we can have the taste that our scheme has similar throughput performance with *IAC*. Although *IAC* needs less time slots in some cases, our scheme outperforms either consumption for less relay node or overall network throughput for more nodes to recover the original data.

So our scheme is applicable to multi-hop multi-source multicast in wireless networks. I will go on with this scheme by analyzing its complexity, achievable throughput and energy consumption.



(a) Example 1 (b) Example 2 (c) Example 3

(d) Example 4 (e) Example 5

Fig. 10: Five example in [3].

E. Practical Issues

Bit-level Synchronization: Like the physical-layer network coding in [1], our work also requires bit-level synchronization for concurrent scheduled neighboring sources or relay nodes. This is already addressed in literature, we can borrow the existing techniques.

Channel State Information: Channel state information is critical for both physical network coding and interference alignment and cancellation. This issue is also discussed in numerous literature. *CSI* can be obtained through training sequences in protocol headers.

References

[1] S. Zhang, S.C. Liew and P.P. Lam, *Hot Topic: Physical-Layer Network Coding*, In proceedings of MOBICOM'06, Los Angeles, CA, USA, Sept. 24-29, 2006.

[2] T. Wang and G.B. Giannakis, *Complex Field Network Coding for Multiuser Cooperative Communications*, In IEEE Journal of Selected Areas in Communications, vol. 26, no. 3, April, 2008.

[3] L.E. Li, R. Alimi, D. Shen, H. Viswanathan and Y.R. Yang, *A General Algorithm for Interference Alignment and Cancellation in Wireless Networks*, In proceedings of INFOCOM'10, San Diego, CA, USA, Mar. 15-19, 2010.