Journal of Communications Vol. 10, No. 1, January 2015

Network Coding Aware QoS Routing for Wireless Sensor Network

Xing Shao[1], Cui-Xiang Wang[1], and Yuan Rao[2]
[1] College of Information Engineering, Yancheng Institute of Technology, Yancheng, 224051, China
[2] College of Information and Computer Science, Anhui Agricultural University, Hefei, 230036, China
Email: {shawxing, ry9925}@gmail.com; wangcuixiang@ycit.edu.cn

Abstract**—**This paper proposes a network coding aware QoS (quality of service) routing (NCQR) for wireless sensor network, which exploits network coding technology to improve the QoS routing of wireless sensor network. Network coding condition with QoS constraint is proposed, which provides proof for coding opportunity detection. To facilitate the evaluation of discovered routes, a novel routing metric, called CQRM (coding aware QoS routing metric), is presented, which jointly considers link quality, node congestion and coding opportunity. Simulation results demonstrate that NCQR decreases the blocking ratio of QoS requests significantly and prolongs network lifetime.

Index Terms**—**Wireless sensor network; QoS; routing algorithm; network coding aware

I. INTRODUCTION

Wireless sensor network [1], [2] is a kind of wireless multi-hop network composed of wireless sensor nodes with constrained power and computing capability. Due to its advantages, wireless sensor network serves a wide range of applications, such as home automation, environment monitoring, and so on.

In recent years, with the emergence of various new applications and rapid development of hardware designing technologies, the quality of service (QoS) for the wireless sensor network is attracting more attention. QoS routing [3]-[5] is an important way to solve this problem.

In 2000, the concept of network coding [6] was proposed, which allows intermediate nodes to code the received packets. Wireless network coding [7], i.e. the network coding in wireless environment, could reduce the number of transmission and save bandwidth. Therefore, it has the potential to improve QoS routing of wireless sensor network.

Based on the combination of network coding and routing algorithm, a few novel routing algorithms for wireless multi-hop network, also known as coding aware routing, have been proposed [8]. The first unicast routing combined network coding, called COPE (coding opportunity), was proposed in [9]. And two typical traffic patterns existing coding opportunity were analyzed. However, coding opportunities in COPE are explored passively in the discovered routes, because COPE separates the process of route discovery and coding opportunity detection, which neglects some potential coding opportunities. To address this problem, Ref. [10] presented network coding and interference aware routing algorithm, which could detect coding opportunities in the route discovery process, thereby further increasing coding opportunities and network throughput. In addition, it could aware the interference to increase network throughput. Moreover, typical traffic patterns in COPE are limited within one hop range. To overcome this limitation, Ref. [11] proposed DCAR (distributed coding aware routing) algorithm, which extends the scope of traffic patterns.

Current routing algorithms combing with network coding mentioned above, mostly focus on increasing coding opportunity to improve network throughput. However, these routings do not consider the characteristic of wireless sensor network, e.g. limited energy equipped sensor nodes. Besides, they do not consider the increasing QoS demand of various newly emerged applications. In addition, the single and excess pursuit of coding opportunity increasing may lead to routes assembling and congestion at nodes with coding opportunities, which degrades QoS performance. Therefore, it is not adequate to exploit these proposed coding aware routings directly to solve the QoS routing problem of wireless sensor network.

Moreover, a few QoS routing algorithms [12]-[14] for wireless sensor network have been proposed currently. Jalel *et al.* [12] proposed an energy efficient and QoS aware multipath routing protocol (EQSR). EQSR maximizes the network lifetime through balancing energy consumption across multiple nodes, and introduces the concept of service differentiation to guarantee the QoS of different traffic. Ref. [13] exploits agents to assist the QoS routing of wireless sensor network. These agents participate in network routing and network maintenance. Besides, particle swarm optimization algorithm is used for the QoS routing optimization. Babar *et al.* [14] proposed energy efficient and QoS aware Routing (EEQR) for clustered wireless sensor network, which determines

---

Manuscript received September 27, 2014; revised January 28, 2015.

This work was supported by the National Natural Science Foundation of P.R. China (No. 61203217, 61170065, 61373017, 61171053 and 61103195), Talents Project of Yancheng Institute of Technology (KJC2014038).

Corresponding author email: shawxing@gmail.com

doi:10.12720/jcm.10.1.24-32

the data prioritization based on message type and content to ensure QoS for different traffic types. And the combination of mobile and static sink is used for data gathering to address energy efficiency and high delay problem. However, these QoS routing algorithms do not exploit network coding to save bandwidth resource and admit more QoS flows into the network. Although network coding need additional energy consumption due to coding/decoding, these additional energy consumption can be neglected compared with the energy saved by network coding [15].

The above two points motivate us to propose a network coding aware QoS routing (NCQR) for wireless sensor network, which could exploit network coding for throughput improvement and provide QoS guaranteed service in delay and bandwidth. The key contributions of this paper include:

1) Coding conditions with QoS constraint in bandwidth, which provide basis for coding opportunity detecting.

2) A novel routing metric, CQRM (coding aware QoS routing metric), which considers link quality, node congestion and coding opportunity.

The rest of this paper is organized as follows. The QoS routing problem formulation are given in section II. Section III describes the design details of NCQR. Simulation results analysis is given in section IV. Section V concludes this paper.

II. PROBLEM FORMULATION

QoS routing with more than one constraint is NP-complete problem [16]. Using heuristic algorithm, multi-constrained QoS routing could work in polynomial time [16]. This paper considers QoS routing with constraints in delay and bandwidth. Before proposing the formulation of QoS routing, some notations are introduced.

A wireless sensor network can be represented as a graph *G=(V,E)*, where *V* and *E* denote the node set and link set of the network respectively. The link from node *i* to its adjacent node *j* is l_{ij}. $cost(l_{ij})$ denotes the link cost of l_{ij} and $band(l_{ij})$ denotes the bandwidth of l_{ij}. Given the source and destination node pair *(S,D)*, *R(S,D)* is the route set from *S* to *D*. *r* denotes one element of *R(S,D)*. *E(r)* is the set of links that *r* traverse.

Suppose $I_{ij}(r)$ is the indicator whether *r* traverses l_{ij}, it can be expressed as (1):

$$I_{ij}(r) = \begin{cases} 1\,, & r\ traverse\ l_{ij} \\ 0\,, & otherwise \end{cases} \qquad (1)$$

Given a QoS request from *S* to *D*, the QoS routing in NCQR can be formulated as a multi-constrained optimization problem:

$$\min_{r\in R(S,D)} Cost(S,D,r) = \min_{r\in R(S,D)} \sum_{i=s}^{D} \sum_{j=s,\, j\neq i}^{D} I_{ij}(r)\,\mathrm{cost}(l_{ij}) \qquad (2)$$

subject to:

(1) $delay(S,D,r) \le Delay$

(2) $band(S,D,r) = \min_{l_{ij}\in r} band(l_{ij}) \ge Band$

where *delay(S,D,r)* denotes the delay of *r*, while *band(S,D,r)* is the minimum bandwidth of links in *E(r)*. *Delay* and *Band* denote the maximum delay and minimum bandwidth of QoS requirements respectively.

III. NETWORK CODING AWARE QOS ROUTING

A. Node Structure

Due to the inherent broadcast and lossy characteristics of the wireless links in wireless sensor network, the link of poor quality is a great challenge to routing algorithm design. If routes traverse through these links of high loss ratio, routing performance will degrade inevitably. To address this problem, each node in NCQR keeps a *NeighborTable* maintaining the quality of links to adjacent nodes and ensures routes traverse through links whose delivery ratio is higher than a threshold *L*. The *NeighborTable* has following items:

- *neighborID*: ID of adjacent node.
- *availBand*: Current available bandwidth at adjacent node.
- *fDeliverRatio*: Delivery ratio of link from current node to adjacent node.
- *bDeliverRatio*: Delivery ratio of link from adjacent node to current node.
- *nnIDs*: The ID list of adjacent node’s adjacent nodes.
- *Status*: The status of adjacent node.

If the *fDeliverRatio* of one adjacent node in *NeighborTable* exceeds *L*, the *status* of the adjacent node is set *Positive*, otherwise *Negative*. To update the *status* of adjacent nodes in time, each node calculates the link delivery ratio periodically as follows.

Each node sends Hello message periodically (period is *τ*) and sets a calculation period *T*. And every node counts the number of Hello messages received from adjacent node in each calculation period and calculates the delivery ratio of corresponding link with interval *T*. Suppose $p_{s,d}^{T}(m)$ denotes the delivery ratio of link from *s* to *d* in the *m*-th calculation period, it is calculated as (3).

$$p_{s,d}^{T}(m) = \frac{Num_{s,d}^{r}(m)}{Num_{s,d}^{t}(m)} = \frac{Num_{s,d}^{r}(m)}{T/\tau} \qquad (3)$$

where $Num_{s,d}^{r}(m)$ represents the number of Hello messages *d* received from *s* in *m*-th calculation period, while $Num_{s,d}^{t}(m)$ denotes the number of Hello messages *s* sent to *d* in the same calculation period.

To save bandwidth resource, the latest information about available bandwidth of current node, the adjacent node list, delivery ratio of links from neighbors to the current node, are piggybacked on the periodical Hello messages.

In addition, for correct decoding of network coding, a circular queue is kept at each node for storage of packets current node sent, received, and opportunistic listened from adjacent nodes. When the queue is full, the latest packet will substitute the oldest one in the queue.

To facilitate coding opportunity detection, a *RelayRouteTable* is necessary as well for storing information of routes traverse through current node, which has following items:

- *sourID*: ID of the current route's source.
- *destID*: ID of the current route's destination.
- *prevhop*: Previous hop of current node in current route.
- *nexthop*: Next hop of current node in current route.
- *reservedBand*: Bandwidth reserved for current route at current node.

If the duration that there is no packet transmitted along one route in *RelayRouteTable* exceeds a timeout period *u*, the route will be cleared from *RelayRouteTable* and the bandwidth reserved for this route in the current node will be released. In NCQR, *u* is set to equal to *Delay*.

The node structure in NCQR is as shown in Fig. 1.

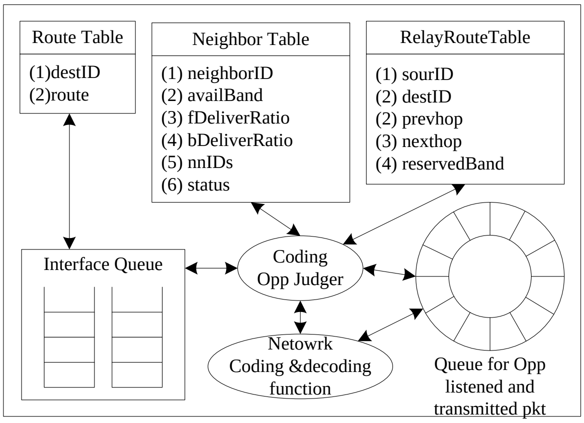

Fig. 1. Node structure in NCQR.

B. Coding Conditions with Bandwidth Constraint

Network coding in NCAR needs to consider the bandwidth constraint of QoS, while network coding in turn affects the bandwidth constraint problem due to its capability of bandwidth saving. Take the scenario in Fig. 2 for example, assume the bandwidth of node *1*, *2*, *C* is *B*. There is a QoS flow (*flow 1*) from *1* to *2* via *C* with bandwidth of B_1 reserved for this flow at *1* and *C* respectively. At this time, a QoS request arrives at node *2*, whose destination is *1* and QoS requirement in bandwidth is B_2.

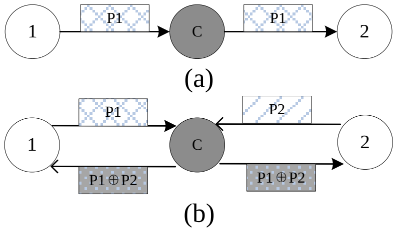

Fig. 2. Node structure in NCQR.

If $B_2>B-B_1$, node *2* using traditional QoS routing will deny the QoS request due to bandwidth shortage. However, according to the typical coding topology in COPE, if the route of the request (*flow 2*) traverses through *C* to *1*, the three nodes (*1,2,C*) and two flows (*flow 1*, *flow 2*) compose the chain topology as in COPE. Considering the bandwidth constraint, as long as $B>max(B_1,B_2)$, packets from *flow 1* and *flow 2* can be coded at *C* satisfying the bandwidth constraint. Then, *C* could admit the QoS request. In NCQR, the coding operation refers to XOR as in COPE.

From Fig. 2, it can be found that the coding opportunity detection should be combined with the bandwidth checking in NCQR, since the node without enough bandwidth, but with coding opportunity (e.g. *C* in Fig. 2), may be one potential and feasible next hop. Bandwidth checking in NCQR is carried out in route request process. Therefore, coding opportunity detection in NCQR should be combined with bandwidth checking in the route request process. Since the coding opportunity detection mechanism in DCAR is in route reply process, it is not suitable for NCQR with bandwidth constraint, although DCAR proposed the general coding condition and extended coding topology range. Before proposing the coding condition with bandwidth constraint, related notations are introduced.

For a route *r*, *r* traverses through node *v* ($v\in r$). The *previous hop node* of *v* in *r* can be expressed using symbol $prev(r,v)$. The *next hop node* of *v* in *r* can be expressed using symbol $next(r,v)$.

Theorem 1 Suppose *n* routes traverse through node *v*, $r_1,r_2,r_3,...r_n$, the corresponding bandwidth constraint of each route is $b_1,b_2,b_3,...b_n$ and the bandwidth at *v* is *B*. $N(v)$ denotes the set of *v*'s neighbors. The necessary and sufficient condition that packets from the *n* routes can be coded at *v* is as follows:

Take any one of the *n* routes, r_i , for any other route r_j ($i\neq j$), the following three items are satisfied.

1) $next(v,r_i)=prev(v,r_j)$ or $next(v,r_i)=N(prev(v,r_j))$;

2) $next(v,r_j)=prev(v,r_i)$ or $next(v,r_j)=N(prev(v,r_i))$;

3) $max(b_1,b_2,b_3,...b_n)\leq B$

Proof:

1) If items (1) and (2) satisfy, r_i and r_j form the typical coding topologie (Chain or "X") or hybrid topology with coding opportunity as in COPE, i.e. r_i could code with r_j at *v*, which means that $next(v,r_i)$ holds packets of r_j ($i\neq j$). Since r_i denotes anyone of the *n* routes, it can be found that for any route, the next-hops of *v* can decode the received coded packets and get their corresponding native packets. Without bandwidth constraint, the packets of the *n* routes could be coded at *v*. Moreover, when packets of multiple routes are coded together, the consumed bandwidth lies on $max(b_1,b_2,b_3,...b_n)$. If item (3) satisfies, the packets of the *n* routes with bandwidth constraint could be coded at *v*. This proves the sufficiency of the coding condition with bandwidth constraint.

2) On the other hand, if packets from the *n* routes can be coded at *v*, it is obvious that for any route the next-hops of *v* can decode the received coded packets, i.e. they hold packets of any other route, which leads to the

satisfaction of item (1) and (2). In addition, the consumed bandwidth of the n routes that could be coded at v is $max(b_1,b_2, b_3,...b_n)$. Obviously, it should be less than total bandwidth B at v, i.e. item (3) satisfies. This proves the necessity of the coding condition with bandwidth constraint.

C. Route Discovery

The route discovery process of NCQR is based on DSR [17] with expansion in network coding aware and QoS guarantee. Suppose source node S receives a QoS request to destination D with QoS requirement in delay and bandwidth. The QoS requirement can be expressed using <*Delay*,*Band*>, where *Delay* and *Band* denote QoS constraints in delay and bandwidth respectively. If no QoS route meets <*Delay*,*Band*> to D in its route table after querying, S initiates QoS route discovery process. Otherwise, S will send packets of the request to D along the available QoS route.

1) The route request process

Step 1.

Node S creates RREQ (Route Request) packet, appending a QoS parameter field to indicate the QoS requirements (<*Delay*,*Band*>). Then S checks whether its available bandwidth is not less than *Band*. If it does not satisfy, the RREQ is dropped. Otherwise, S picks the nodes with *Positive* status in its *NeighborTable* to compose a valid next hop set *valNeSet*. Then the *valNeSet* is inserted into the RREQ, and RREQ is forwarded.

Step 2.

Step 2.1: Upon receiving RREQ, the intermediate node checks the following items:

1) Whether it is not in the *valNeSet*;

2) Whether the transmission time experienced by RREQ from its creation already exceeds *Delay*;

3) Whether the number of hops experienced by RREQ is greater than *TTL* in RREQ;

4) Whether the intermediate node already presents in the path traversed by RREQ (avoid looping).

If anyone of the above four items satisfies, the RREQ is dropped. Otherwise, turn to Step 2.2.

Step 2.2: Intermediate node checks whether itself is the destination node. If so, the path in RREQ is copied to create RREP packet, the RREQ is dropped, and the destination node initiates route reply process. Otherwise turn to Step 2.3.

Step 2.3: The intermediate node checks whether it is marked as existing coding opportunity in *valNeSet*. If so, the intermediate node marks the last hop node in path in RREQ as existing coding opportunity and appends the number of routes participate in coding. Turn to Step 2.4

Step 2.4: The intermediate node initiates coding aware process with bandwidth constraint as shown in TABLE I and stores the *Result*.

Step 2.5: The intermediate node checks whether its available bandwidth is less than *Band*.

1) If its available bandwidth is less than *Band* and the set *Result* is empty, discard the RREQ;

2) If its available bandwidth is less than *Band* and the set *Result* is not empty, take the next hop nodes in *Result* to constitute a new *valNeSet*, and corresponding number of routes participate in coding is recorded in *valNeSet* as well, then replace the *valNeSet* in RREQ with the new *valNeSet* and turn to Step 2.6;

3) If its available bandwidth is greater than *Band* and the set *Result* is empty, the intermediate node's bandwidth of *Band* size is *temporarily* reserved. If the intermediate node does not receive the RREP in time W, the reserved bandwidth will be released (In NCQR, W is set to twice of *Delay*). Then the intermediate node picks the nodes with *Positive* status in its *NeighborTable* (exclude exclude the last hop of RREQ) to compose a new *valNeSet* to replace the old one in RREQ. Turn to Step 2.6.

4) If its available bandwidth is greater than *Band* and the set *Result* is not empty, further check whether the number of its *Positive* adjacent nodes (exclude the last hop of RREQ) is greater than the number of adjacent nodes in *Result*. If so, the intermediate node's bandwidth of *Band* size is *temporarily* reserved. Otherwise, bandwidth reservation is not necessary. Then the intermediate node picks the nodes with *Positive* status in its *NeighborTable* (exclude the last hop of RREQ) to compose a new *valNeSet* to replace the old one in RREQ. And according to *Result*, the node in *valNeSet* and also in *Result*, is marked as existing coding opportunity and appended with the number of routes participate in coding. Turn to Step 2.6.

Step 2.6: Append the ID of the intermediate node with its calculated congestion index into the path field in RREQ and update the *TTL* value, then forward the RREQ. Turn to Step2.1.

TABLE I. CODING AWARE PROCESS WITH BANDWIDTH CONSTRAINT

```
Coding Aware Process with bandwidth constraint
Input: node $v$, route of RREQ $r_d$, the Positive neighbors of $v$ (exclude the last hop of RREQ), routes (without coding at $v$) that traverse through $v$: $r_1,r_2,r_3,...r_n$
Output: Result, whose element is tuples of next hop that let $r_d$ exist coding opportunity at $v$ and number of routes participate in coding;
$Result \leftarrow \varnothing$;
For $v$'s each Positive neighbor $n_i$, $i=1$ to $m$ ($m$ is the number of Positive neighbors of $v$ exclude the last hop of RREQ) do
    $r_d$ appended with $n_i$ composing new route $r'_d$;
    $CodeSet \leftarrow \{r'_d\}$;
    For $r_j$, $j=1$ to $n$ do
        If $r_j$ meets the coding conditions with routes in CodeSet then
            $CodeSet=CodeSet \cup \{r_j\}$;
    Endfor
    $Num=|CodeSet|$;  //Num is the number of items in CodeSet
    If $Num \neq 1$ then
        $Result=Result \cup \{<n_j, CodeSet, Num>\}$;
Endfor
return Result
```

2) Route reply process

Having received K_q RREQs satisfying delay constraint, or having waited for time of (*Delay*-d) (d is the delay experienced by first arrival RREQ) after the first arrival of RREQ and received N_q ($1 \leq N_q \leq K_q$) RREQs, the destination node *D* initiates the route reply process.

Step 1: Then *D* evaluates the cost of path in each RREQ based on CQRM described in section III.D, and selects the path with the minimum cost as the route to create RREP. Then RREP is forwarded along the reverse path to *S*.

Step 2: Upon receiving a RREP, intermediate node will transfer the *temporarily* reserved bandwidth for the route in RREP, if exists, to be *officially* reserved bandwidth.

Step 3: Upon receiving RREP, *S* establishes QoS route to *D* in route table based on the path in RREP and begins to send packets along the route. If *S* does not receive any RREP in time T_s, it means there is not any QoS route from *S* to *D* currently in the network, the route discovery process fails, and the QoS request is denied. In NCQR, T_s is set to three times of *Delay*.

D. Coding Aware QoS Routing Metric (CQRM)

Through route discovery process, the destination node may eventually get multiple routes meeting QoS requirements in delay and bandwidth. It is difficult and unrealistic to find a route with minimum delay, surplus bandwidth, and maximum coding opportunities. The route with maximum coding opportunities may own large delay, while route with small delay may traverse through congestion node. Therefore, selecting an appropriate route metric, which reflects the link quality, node congestion and coding opportunity, to evaluate paths discovered by RREQ is indispensable. Considering above factors, this paper proposes a novel routing metric, CQRM (coding aware QoS routing metric). Before the definition of CQRM, several related definitions are introduced.

Definition 1 Time before Congestion

Suppose *I* and *O* denote the receiving rate and sending rate at node *i* respectively, Q_l is the queue length, *q(t)* denotes the number of packets in the queue, the *Time before Congestion* of node *i* at time *t* can be expressed as (4) :

$$\xi_i(t) = \begin{cases} \dfrac{(Q_l - q(t)) \times P_{avg}}{I - O}, & I > O \\ \infty, & I = O \end{cases} \tag{4}$$

where $q(t) = \dfrac{1}{\Delta t}\int_{t-\Delta t}^{t} q(i) \times d_i$. This parameter represents the left time before congestion occurs, which reflects the load degree at a node on the other hand.

Definition 2 Node Congestion Index

Node Congestion Index reflects the congestion extent at node *i*, and it is expressed as (5):

$$\delta_i(t) = \begin{cases} e^{1/\xi_i(t)}, & I > O \\ 1, & I = O \end{cases} \tag{5}$$

The parameter Node Congestion Index is the normalized Time before Congestion and reflects the congestion degree of nodes.

Definition 3 Resource Consumption Index

Resource Consumption Index *γ* denotes the extent of resource saving using network coding in a transmission. If packets of current route could be coded with packets of other *(m-1)* routes (*m>1*), then *γ=1/m*. Otherwise, *γ=1*. The value *m* can be obtained from the *Result* of the coding aware process as in TABLE I. The calculation of *γ* is as (6):

$$\gamma = \begin{cases} 1\ , & \textit{without coding opportunity} \\ 1/m, & \textit{coded with packets of } (m-1) \textit{ routes} \end{cases} \tag{6}$$

The parameter Node Congestion Index is the reciprocal of the number of coded flows indeed.

Definition 4 Coding Gain Factor

Coding Gain Factor *λ* reflects the contribution of coding opportunity to route for resource saving. It can be expressed as (7):

$$\lambda = e^{\gamma - 1} \tag{7}$$

The parameter Coding Gain Factor is the normalized Resource Consumption Index and reflects the contribution of network coding to network.

Definition 5 Coding aware QoS Routing Metric (CQRM)

The *CQRM* value of a route is defined as follows:

$$CQRM = \sum_{i=1}^{H} ETX_i \times \delta_i \times \lambda_i = \sum_{i=1}^{H} ETX_i \times \delta_i \times e^{\gamma_i - 1} \tag{8}$$

where *ETX* [18] indicates expectation number of transmission and *H* is the number of links in the route.

According to the CQRM definition as (8), *ETX* reflects link quality, δ_i reflects the degree of node congestion, and λ_i reflects the contribution of coding opportunity to route for bandwidth saving. Therefore, CQRM could reflect link quality, node congestion and coding opportunity.

Theorem 2 Node congestion index and coding gain factor could guide NCQR in favor of paths whose nodes have light load and more coding opportunities.

Proof: If a node of route has light load, its congestion index is small according to (5), and less likely to be congested. As shown in (7), it can be found that $\lambda = \begin{cases} 1\ , & \gamma = 1 \\ e^{\gamma-1} < 1, & \gamma < 1 \end{cases}$. If there exists coding opportunity at a node in the route, its resource consumption index is less than 1, leading to coding gain factor less than 1. However, if there exists no coding opportunity, the coding gain factor is 1. Smaller is the *γ*, smaller the *λ*. Therefore, the path whose nodes have light load and more coding opportunities has smaller CQRM value and greater probability of being chosen as route by NCQR.

E. Overhead and Complexity Analysis

As a distributed routing algorithm, the overhead of NCQR mainly involves following two parts:

1. Periodical Hello packets for calculating link delivery ratio.

2. Flooding of RREQ packets in route discovery process.

Actually, some essential information for route discovery and coding aware process is piggybacked on the periodical Hello messages as explained in section III.A. And the overhead of RREQ packets is associated with any on-demand routing like DSR[16] or AODV[18]. Therefore, NCQR does not increase overhead significantly compared with other coding aware routing algorithms (e.g. DCAR) and QoS routing algorithms (e.g. QUORUM).

Theorem 3 The storage complexity of NCQR is *O(R)*

Proof: As shown in Fig. 1, each node in NCQR owns a circular queue for decoding, a *RouteTable* toward other nodes, a *NeighborTable* about its neighbors and a *RelayRouteTable* about the routes traverse itself. The storage complexity of the circular queue and *RouteTable* towards one node is *O(1)*. The total number of other nodes (excluding itself) is $|V|-1$. Thereby the storage complexity of *RouteTable* is *O(1)*. The storage complexity of *NeighborTable* about one neighbor and *RelayRouteTable* about one route is *O(1)*. Suppose N is the maximum number of neighbors and R is the maximum number of relayed routes at a node. It is obvious that $N\leq|V|-1$. The storage complexity of *NeighborTable* is *O(1)*, while that of *RelayRouteTable* is *O(R)*. Therefore, the total storage complexity of NCQR is *O(R)*.

Theorem 4 The computation complexity of NCQR is $O(PR+HK_q(|V|-1))$

Proof: The main computation in NCQR is the calculation in the coding aware process as in Table I and CQRM value calculation of paths at destination. Suppose P is the maximum number of *Positive* neighbor of a node, the computation complexity of coding aware process is *O(PR)* according to Table I.

Suppose H is the maximum number of hops of a route. Since the CQRM value calculation of a hop only involves simple mathematical operation, whose computation complexity is *O(1)*, the computation complexity of a route's CQRM value calculation is *O(H)*. And the maximum number of RREQ received from a certain source node (excluding itself) is K_q ($K_q<<|V|-1$). The maximum number of RREQ received from all source nodes, i.e. the maximum number of route CQRM value calculation is $K_q\times(|V|-1)$. The total computation complexity of CQRM value calculation is $O(HK_q(|V|-1))$. Therefore, the total computation complexity of NCQR is $O(PR+HK_q(|V|-1))$.

IV. PERFORMANCE EVALUATION

A. Simulation Parameters

In order to verify the performance of NCQR, simulations are carried out using NS2 [19]. To analyze the network coding aware capability of NCQR, NCQR is compared with DCAR, which is a classical coding aware routing. NCQR is also compared with EQSR to investigate its ability in QoS service providing. NCQR and EQSR provided QoS guaranteed service, while DCAR provide best effort service and admit all flows into network. The network topology consists of 200 nodes randomly placed in an area of 600m×600m. NCQR is implemented on top of 802.15.4 MAC with channel bandwidth of 250 Kbps at each node. To investigate the energy consumption of NCQR, the energy model presented in [20] is exploited in the simulations and initial energy of each node is 100J.

TABLE II. SIMULATION PARAMETERS

Parameter	Value
Transmission Range	50m
Interference Range	100m
Work Mode	Promiscuous Mode
Broadcast Mode	Pseudo-broadcast
Simulation time	500s
Queue Length	100 packets
Queue Type	FIFO

Each node broadcasts Hello messages every 100ms ($\tau=100ms$), and calculates the link delivery ratio once every second ($T=1s$). The threshold of link delivery ratio L is set to 0.7. And the maximum number of RREQ received from a source node K_q is set to 4.

Besides, QoS traffic is sent as CBR, whose QoS requirement in delay and bandwidth is 100 ms ($Delay=100ms$) and 5 Kbps ($Band=5Kbps$). The duration of each flow is 2 minutes. All QoS flows are of identical traffic characteristics, i.e. data rate, packet size, and QoS requirement in simulation. The source and destination of each QoS flow is randomly selected from the 20 nodes. In addition, simulations under different QoS request arrival rate is carried out. For the sake of fairness, each simulation is run 10 times, and the average of 10 simulations results is taken as final result. The other detailed simulation parameters are presented in Table II.

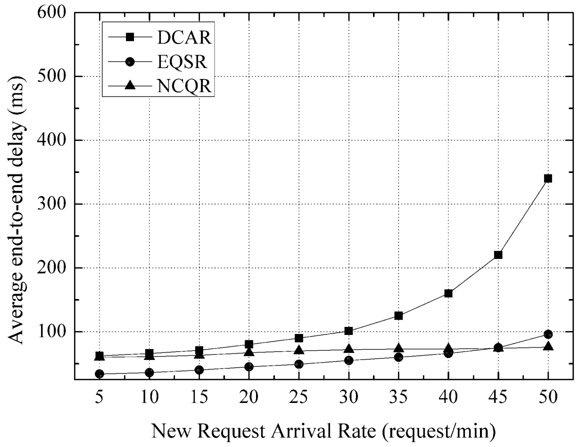

Fig. 3. Average end-to-end delay.

B. Simulation Results Analysis

Fig. 3 graphs the transition of average end-to-end delay with the increasing of new request arrival rate. It is obvious from the figure that the average end-to-end delay of NCQR is almost same as DCAR, and slightly higher than EQSR when the new request arrival rate is less than 15 request/min. However, when the new request arrival rate is larger than 20 request/min, the average end to end delay of NCQR is lower than DCAR and close to EQSR. When the new request arrival rate exceeds 45 request/min, the average end to end delay of NCQR is even lower than EQSR.

This is reasonable because that NCQR and DCAR need to consider coding opportunities which is usually not the shortest routes as EQSR. Therefore, the average end-to-end delay of NCQR and DCAR is close, when the new request arrival rate is low. With the increasing of new request arrival rate, congestion occurs in the networks using DCAR resulting in significant increasing of delay, since all flows are admitted into networks. Due to the ability of QoS guaranteeing, NCQR and EQSR could ensure the delay of admitted flows within Delay. Besides, for exploiting coding opportunity, NCQR could save bandwidth resource and alleviate network load, which leads to its delay lower than EQSR in case of high new request arrival rate.

Fig. 4 presents the throughput versus new request arrival rate. It can be seen from this figure that when the new request arrival rate is less than 15 request/min, the throughput of the three algorithms is very close. When the new request arrival rate exceeds 25 request/min, NCQR is lower than DCAR, and is slightly higher than EQSR.

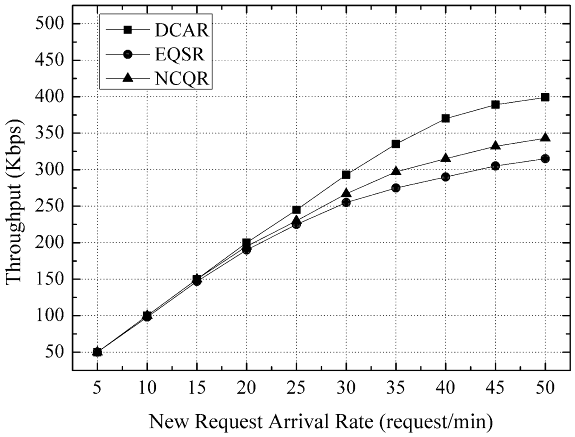

Fig. 4. Throughput.

The reason is that when the new request arrival rate is low, NCQR and EQSR could admit almost all requests. With the increasing of new request arrival rate, NCQR and EQSR begin to reject some requests resulting in throughput decreasing compared with DCAR. Whereas, due to the capability of coding aware, NCAR could alleviate network load and admit more requests into network resulting in the throughput improvement compared with EQSR.

Fig. 5 depicts the average energy consumption versus new request arrival rate. When the new request arrival rate is lower than 15 request/min, the energy consumption of NCQR is close to DCAR and slightly higher than EQSR, since the coding opportunity is fewer and the route of EQSR is optimal.

With the increasing of new request arrival rate, the energy consumption of NCQR grows more slowly compared with DCAR and EQSR. The reason is that, NCQR could exploit network coding to reduce the transmission number and energy consumption. However, since DCAR admit all flows into network, the congestion occurs with the increasing of energy consumption.

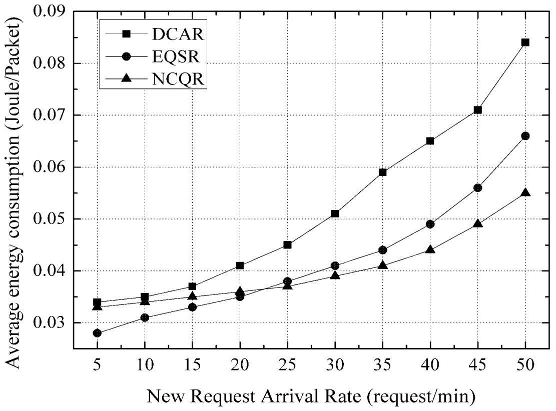

Fig. 5. Average energy consumption.

Fig. 6 graphs the network lifetime of the three routings under different new request arrival rate. It is obvious from Fig. 6 that NCQR is slightly higher than EQSR and higher than DCAR. And the gap between NCQR grows with the increasing of new request arrival rate. The reason is that NCQR exploit network coding to reduce the node energy consumption, while EQSR could not exploit this advantage, and the number of coding opportunities grows with the increasing of new request arrival rate.

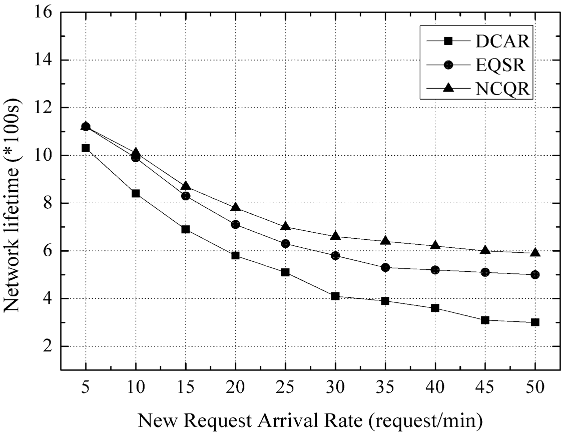

Fig. 6. Network lifetime.

Fig. 7 illustrates packet delivery ratio under different new request arrival rate. It is clear from Fig. 7 that, DCAR outperforms than other two routings, since DCAR admits all flows, while EQSR and NCQR need to provide QoS and admit certain number of flows according to network condition. Besides, NCQR is higher than EQSR. The reason is that NCQR exploit network coding and admit more flows into network.

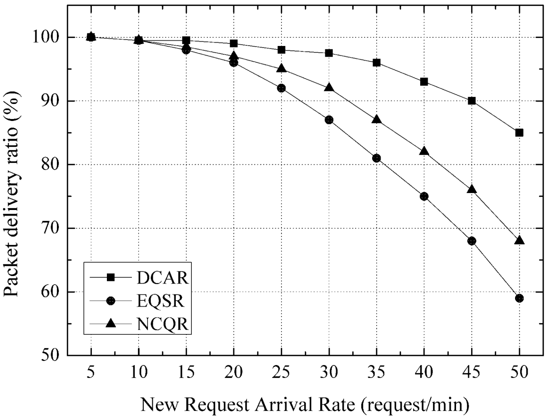

Fig. 7. Packet delivery ratio.

Fig. 8 presents the request blocking ratio versus new request arrival rate. Since DCAR admit all flows into network, the request blocking ratio of DCAR is always 0.

As shown in Fig. 8, when the new request arrival rate is less than 20 request/min, the request blocking ratio of NCQR and EQSR is very close. When the new request arrival rate exceeds 20 request/min, the request blocking ratio of NCQR is average 7 percentage points lower than EQSR. The striking results of the NCQR are ascribed to its ability of network coding aware in routes which lead to bandwidth saving to admit much more QoS requests into network compared with EQSR, especially in the case of high new request arrival rate.

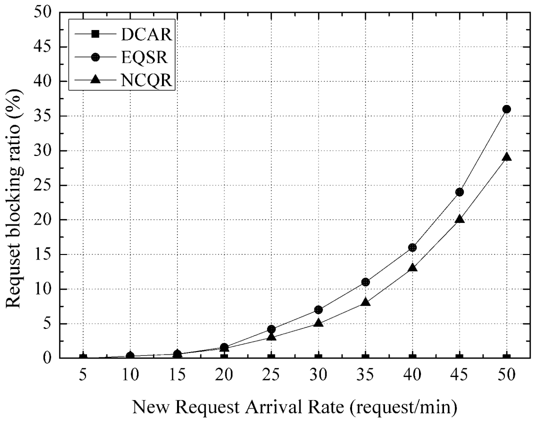

Fig. 8. Request blocking ratio.

V. CONCULSION

In this paper, we presented a distributed routing algorithm, network coding aware QoS routing (NCQR), for wireless sensor network. In addition, this paper proposed the coding condition with bandwidth constraint and a novel routing metric CQRM which considers link quality, node congestion and coding opportunity. The routing performance has been evaluated in terms of average end-to-end delay, throughput, average energy consumption, network lifetime, packet delivery ratio and request blocking ratio. Coding conditions with bandwidth constraint, computational and storage complexities of NCQR have been proved.

To sum up, NCQR could provide QoS guaranteed service in delay and bandwidth, admit more QoS flows into wireless sensor network, and prolong the network lifetime with some expense in average end-to-end delay, coding opportunity detection and coding operation. In other words, NCQR is a desirable combination of QoS guaranteed service and network coding. This fact renders it an excellent choice for wireless sensor networks. Future work will include enhancing NCQR to deal with network coding with rate adaptation.

ACKNOWLEDGMENT

The authors wish to thank anonymous reviewers for their valuable comments and suggestions for the improvement of this paper. This work was supported in part by the National Natural Science Foundation of P.R. China (No. 61203217, 61170065, 61373017, 61171053 and 61103195), Talents Project of Yancheng Institute of Technology (KJC2014038), Cooperative Innovation Fund Project of Jiangsu Province (BY2014108-29).

REFERENCES

[1] I. F. Akyildiz, S. Weilian, Y. Sankarasubramaniam, and E. Cayirci, “A survey on sensor networks,” *IEEE Communication Magazine,* vol. 40, no. 8, pp. 102-114, July 2002.

[2] J. Yick, B. Mukherjee, and D. Ghosal, “Wireless sensor network survey,” *Computer Networks*, USA, vol. 52, pp. 2292-2330, December 2008.

[3] K. Akkaya and M. Younis, “A survey on routing protocols for wireless sensor networks,” *Journal of Ad Hoc Networks*, vol. 3, no. 3, pp. 325-349, March 2005.

[4] R. Sumathi and M. G. Srinivas, “A survey of QoS based routing protocols for wireless sensor networks,” *Journal of Information Process System*, vol. 8, no. 4, pp. 589-602, June 2012.

[5] D. Chen and P. K. Varshney, “QoS support in wireless sensor networks: A survey,” in *Proc. International Conference on Wireless Networks, ICWN-2004*, Las Vegas, Nevada, USA, June 2004, pp. 227-233.

[6] R. Ahlswede, N. Cai, S. Y. Li, and R. W. Yeung, “Network information flow,” *IEEE/ACM Transactions on Information Theory*, USA, vol. 46, pp. 1204-1216, April 2000.

[7] C. Fragouli, D. Katabi, A. Markopoulou, M. Medard, and H. Rahul, “Wireless network coding: Opportunities and challenges,” in *Proc. Military Communications Conference*, 2007, 29-31 Oct. 2007.

[8] M. A. Iqbal, B. Dai, B. Huang, A. Hassan, and S. Yu. “Survey of network coding-aware routing protocols in wireless networks,” *Journal of Network and Computer Applications*, vol. 34, no. 6, pp. 1956-1970, June 2011.

[9] S. Katti, H. Rahul, W. J. Hu, D. Katabi, M. Mdard, and J. Crowcroft, “XORs in the air: Practical wireless network coding,” *IEEE Transactions on Networking*, vol. 16, pp. 497-510, March 2008.

[10] R. H. Hou, S. K. Qu, K. S. Lui, and J. D. Li, “Coding and Interference aware rouing protocol in wireless networks,” *Computer Communications*, vol. 36, no. 17, pp. 1745-1753, November 2013.

[11] L. Jilin, C. S. Lui John, and D. M. Chiu, “DCAR: Distributed coding-aware routing in wireless networks,” *IEEE Transactions on Mobile Computing*, vol. 9, pp. 596-608, April 2010.

[12] J. Ben-Othman and B. Yahya, “Energy efficient and QoS based routing protocol for wireless sensor networks,” *Journal of Parallel and Distributed Computing*, vol. 70, no. 8, pp. 849-857, August 2010.

[13] M. Liu, S. J. Xu, and S. Y. Sun, “An agent-assisted QoS-based routing algorithm for wireless sensor networks,” *Journal of Networks and Computer Applications*, vol. 35, pp. 29-36, September 2012.
[14] B. Nazir and H. Hasbullah, “Energy efficient and QoS aware routing protocol for clustered wireless sensor network,” *Computers and Electrical Engineering*, vol. 39, pp. 2425-2441, February 2013
[15] D. Estrin, “Wireless sensor networks tutorial Part IV: Sensor network protocols,” in *Proce. Mobicom*, USA, 2002, pp. 23-28.
[16] F. A. Kuipers and P. F. A. van Mieghem, “Conditions that impact the complexity of QoS routing,” *IEEE/ACM Transactions on Network*, USA, vol. 13, no. 4, pp. 717-730, April 2005.
[17] D. B. Johnson, D. A. Maltz, and Y. C. Hu. (February 2007). The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4. IETF. RFC 4278. [Online]. Available: http://www.rfc-editor.org/rfc/rfc4728.txt
[18] D. S. J. De Couto, D. Aguayo, J. Bicket, and R. Morris, “A high-throughput path metric for multi-hop wireless routing,” *Wireless Networks*, vol. 11, pp. 419-434, April 2005.
[19] The network simulator: Ns2. (2011). [Online]. Available: http://www.isi.edu/nsnam/ ns/
[20] S. D. Muruganathan, D. C. F. Ma, R. I. Bhasin, and A. O. Fapojuwo, “A centralized energy-efficient routing protocol for wireless sensor networks,” *IEEE Communication Magazine*, vol. 43, no. 3, pp. S8-S13, 2005.

Xing Shao born in 1985, he received his B.S. in Computer Science and Technology and Ph.D. in Information Network from Nanjing University of Posts and Telecommunications in 2007 and 2013 respectively. He is currently a lecturer in the college of information engineering at Yancheng Institute of Technology, China, where he is engaged in teaching and research on Internet of Things. His research interests include wireless sensor network, routing algorithm, network coding, etc.

Cui-Xiang Wang born in 1982, she received her B.S. in Computer Science and Technology form Qufu Normal University in 2007, and M.S. in Computer Software Theory from Nanjing University of Posts and Telecommunications in 2010. She is currently an assistant in the college of information engineering at Yancheng Institute of Technology, China, where she is engaged in teaching and research on Internet of Things. Her research interests include wireless sensor network, routing algorithm, network security, etc.

Yuan Rao born in 1982, he receives a Diploma and a Ph.D. in the college of computer from Nanjing University of Posts & Telecommunications. He is currently a lecturer in the college of information and computer science at Anhui Agricultural University, China, where he is engaged in teaching and research on network engineering. He has served as a reviewer in several international journals and conferences relevant to information network and evolution computation. His research interests include intelligent agents, evolution computation, satellite routing, agricultural informationization and sensor network, etc.