Congestion control mechanism using Traffic Aware Dynamic Routing (TADR) algorithm

S.Pavithra, R.Bavani, J.Uma Sawmiya, Dr.K.Vimaladevi

B.E. IV Year Department of Computer Science and engineering, Velammal Engineering College, Chennai.

Professor, Department of Computer Science and engineering, Velammal Engineering College, Chennai.

Abstract: Traffic-Aware Dynamic Routing (TADR) algorithm is proposed to route the packets around congestion areas and scatter the excessive packets along multiple paths consisting of idle and under loaded nodes. Utilizing the concept of potential in classical physics, TADR algorithm is designed for constructing hybrid virtual potential field using depth and normalized queue length to force the packets to steer clear of obstacles.

Keyword- Dynamic routing, Congestion, Hybrid potential field, idle nodes, under loaded nodes.

I. INTRODUCTION

The congestion problem in Wireless Sensor Networks (WSNs) is quite different from that in traditional networks. Most current congestion control algorithms try to alleviate congestion by reducing the rate at which source nodes inject packets into the networks. This traffic control scheme always decreases the throughput. Hence it has to violate the fidelity level required by applications. Therefore, we present a solution that sufficiently exerts idle or under loaded nodes to alleviate congestion and improve overall throughput in WSNs. To achieve this goal, TADR algorithm is proposed where it identifies the congestion areas and then divides the packets to be sent along multiple paths, and finally reaches the destination. This algorithm is designed through constructing hybrid potential field using depth and normalized queue length to force the packets to steer clear of obstacles.

II. HISTORY

Congestion in WSNs has negative impact on performance, namely, decreased throughput and increased per packet energy consumption. Due to the centralized traffic pattern in WSN, just bypassing the hot spots is ineffective to eliminate congestion because it will reappear near sink. For example, the data generated during crisis state are of utmost important and loss of such data can violate the purpose of deploying unattended sensor network.

In other words, congestion control in WSNs must not only be based on network capacity but also on fidelity required by applications. Most of the prior works basically try to throttle the incoming traffic into the network once congestion is detected. Although traffic control strategies are effective to alleviate congestion in traditional networks, they are restricted for the following reason: “Reducing source traffic during crisis state is undesirable since it will significantly violate fidelity requirements.”
It may be a better option to increase capacity by turning on more resources to accommodate excessive incoming traffic during crisis state. There are various congestion control schemes such as capacity planning, end-to-end or hop-by-hop traffic control connection admission control and buffering. It has been found that the selection of congestion control schemes should depend upon the characteristic of congestion.

ii. Path prediction and packet splitting:
Here the path for the transaction of the file is detected and the data is divided into number of packets depending on the size of data. In highly loaded nodes, the surface of the "bowl" is smooth and hence this algorithm acts as shortest path routing.

iii. Packet receiving and joining:
A typical routing loop is caused by local minimal potential, which is a hollow in our bowl model. At the beginning, the nodes around this minimal potential node may send their packets to it, so this hollow will be filled up after sometime. Once the potential of this node goes higher than that of any node around it, the node will send back packets.
IV. TADR ALGORITHM:

TADR- update message from processing:
If received an updatemessage(u_msg) from a
neighbouring node(neighbor_id)
1: insertToRoutingTable(u_msg,neighbor_id)
2: for each entry in RoutingTable
   3: w=id of the neighbor;
   4: c=cost of radio link to w;
   5: d=depth of w;
   6: q=queue length of w;
   7: Fd(w)=(Local_depth-d)/w;
   8: Fq(w)=(Local_queuelength-q)/c;
   9: Fm(w)=(1-α)Fd(w)+ α Fq(w) ;
10:end for;
Recalculate the depth:
11: Select the lowest depth from the RoutingTable as
12: SetLocalDepth(LD+1);
Choose the nexthop node:
13: Selects from the entries with QL<1//RULE1 according to max-Fm, max-Vm, min-depth, min-cost, Random in turn
TADR-Time to update:
If one those events occurred:
1: Timeout of the most updating interval
2: Depth changed
3: The variation of Queue Length exceeds Qupdate_threshold
   1: if(Timeout of the least update interval)then
   2: sendupdatemsg();
   3: else
   4: updatemsgPending=TRUE;
   5: endif;
Processing of update message:

When a node receives update message from one of its
neighbors, it will refresh its routing table and reselect
nexthop node according to the algorithm. TADR uses the
steepest gradient method to choose its parent. Most
precisely, if there are more than one neighbor that has
maximum force Fm, TADR chooses the nexthop node
according to the maximum potential Vm, minimum depth
of neighbors and minimum cost of links. In case id TADR
still cannot determine its parent, then it will choose one
node randomly.
V. ADVANTAGES:
- Faster transfer of data
- Traffic is very low.
- Very low packet loss
- Efficient use of bandwidth
- Promotes security to data.

VI. CONCLUSION
In this paper, we have presented an improvement of traffic aware dynamic source routing protocol by proposing a new metric to evaluate routes. This metric is based on nodes weight computed by combining two parameters which are the power of node and its stability. These are assumed to be the most important parameters in choosing the routes. Then using these weights, we can choose the best route that may be a long one, but that is the best route according to our proposed system. Whenever two routes have near values of weights, we can choose the one with minimum number of hops.

REFERENCES


