IJCSN International Journal of Computer Science and Network, Volume 4, Issue 4, August 2015
ISSN (Online) : 2277-5420 www.IJCSN.org
Impact Factor: 0.417

A Survey on Congestion Control Routing for Wireless Network

[1] B. Rajasekaran [2] Dr. C. Arun

[1] Research scholar, Dept of ECE, St Peters University,
Chennai, Tamilnadu, India

[2] Professor, Dept of ECE, R.M.K. College of Engineering & Technology,
Chennai Tamilnadu, India

Abstract - Wireless networks have long suffered from congestion. It occurs when source node transmits the data at a rate greater than that of a network device can accommodate. It causes the buffer on such devices to fill up and possibly overflow. It results in loss of packets. Most of the Routing protocols which are currently in use are not congestion Adaptive. Existing congestion control schemes treat congestion as an individual problem and propose ad hoc solution that is dissatisfied. Routing should not be aware of, but also be adaptive to network congestion. Adaptation to the congestion helps to increase both the effectiveness and efficiency of routing. These problems are solved by the congestion-aware routing protocols in certain degree. These protocols which are adaptive to congestion status of mobile ad-hoc network can greatly improve the network performance. Further it allows a network to operate in the region of low delay and high throughput. This survey paper concentrates on congestion detection and congestion avoidance.

Keywords – *Congestion Control, Transmission Control Protocol, Active Queue Management, Congestion Avoidance.*

1. Introduction

Congestion when the total demands for resources exceeds the total available resources[1] .In other words we can say a network is in a state of congestion when the quality of the service delivered to the user decreases, whenever there is increase of load in a network. Take an example of downloading a 1 GB file .When there is no congestion, the file gets downloaded in few minutes, but on day when there is congestion in a network same file gets downloaded in hours. The problem of congestion has become an important issue these days.

1.1. Effects of Congestion

Congestion affects two vital parameters of the network performance, namely throughput and delay. The throughput can be defined as the percentage utilization of the network capacity. Throughput is affected as offered load increases. Initially throughput increases linearly with offered load, because utilization of the network increases. However, as the offered load increases beyond certain limit, say 60% of the capacity of the network, the throughput drops. If the offered load increases further, a point is reached when not a single packet is delivered to any destination, which is commonly known as deadlock situation. The ideal one corresponds to the situation when all the packet introduced are delivered to their destination up to the maximum capacity of the network. The second one corresponds to the situation when there is no congestion control. The third one is the case when some congestion control techniques are used. These prevent the throughput collapse, but provides lesser throughput than the ideal condition due to overhead of the congestion control technique[2].

2. Congestion Detection for Wireless Sensor Network

The major issue in wireless sensor network is congestion. So accurate and efficient congestion detection plays an important role in WSN. There are various detection techniques which have low cost in terms of energy and computation complexity. Those techniques are discussed below:

Energy efficient congestion detection method like CODA deals with various degrees of congestion depending on the sensing application [3]. Congestion detection protocols can be implemented by two main schema i) Open-loop hop-by-hop backpressure technique, ii) Closed-loop multi-source regulation technique. In Open-loop hop-by-hop backpressure technique, backpressure is generated as long as congestion is detected when an upstream node (toward the source) receives a backpressure message it decides whether or not to further propagate the backpressure upstream based on its own local network conditions. In closed loop multi-source regulation technique, when the source event rate is less than some fraction of the maximum theoretical throughput of the channel, the source regulates itself. However a source is more likely to contribute to congestion and therefore closed-loop congestion control is triggered, the source only enters sink regulation if this threshold is exceeded. At this point a source requires constant, slow time-scale feedback (e.g., ACK) from the sink to maintain its rate. The reception of ACKs at sources are served as a self-clocking mechanism allowing sources to maintain their current event rates. In contrast, it fails to receive ACKs which forces a source to reduce its own rate.

In Hop-by-hop Backpressure [4] if the sink is congested, backpressure spatially spreads the congestion and helps alleviate congestion quickly. In addition, hop by-bop control supports in-network data processing. Once congestion is detected, the receiver will broadcast a suppression message to its neighbors. Hop-by-hop control signals are propagated upstream toward the source. It needs at least one round-trip-time (RTT) to detect congestion. The hop-by-hop backpressure can immediately response to the congestion at the intermediate node without incurring the round trip delay that reduces feedback's effectiveness. Queue Occupancy [5] is a simple way to detect congestion relies on monitoring a sensor's queue size: if the fraction of space available in the output queue falls below a high water mark á , then congestion bit of outgoing packets is set. Otherwise the congestion bit is cleared.

In Receiver-based Congestion Detection [3], it use a combination of the present and past channel loading conditions and the current buffer occupancy, to infer accurate detection of congestion at each receiver side. Once congestion is detected, nodes signal their upstream neighbors via a backpressure mechanism. In Event to Sink Reliable Transport [6], a sensor sets a congestion notification bit in the packet header if its buffer is full. The sink periodically computes a new reporting rate based on a reliability measurement, the received congestion notification bits and the previous reporting rate. In Congestion Control and Fairness [7], it uses packet service time to deduce the available service rate and detects congestion. Each Sensor node uses rate adjustment based on its available service rate and number of child nodes. CCF provides simple fairness for all nodes with same throughput. But fairness can maintained while each node gets same priority.

Intelligence Congestion Detection [8] method is used to measure local congestion level at each intermediate node, the packet inter-arrival time (Pa) and packet service time(Pb) at MAC layer is taken into consideration. Using Pa and Pb we can derive Congestion Degree (CD)[7][9]. By taking the value of CD into consideration we can detect the occurrence of congestion.

$$CD = Pb/Pa$$

If the value of CD is greater than 1 we can ensure that there is no congestion occurred, if the value results in lesser than 1 the congestion is been detected.

3. Congestion Control Schemes

Basic congestion control schemes

- Slow start
- Fast retransmission and Fast Recovery(Reno)

3.1. Slow Start

Slow start reduces the burst affect when a host first transmits. It requires a host to start its transmissions slowly and then build up to the point where congestion starts to occur[5].The host does not initially know how many packets it can send, so it uses slow start as a way to gauge the network's capacity. A host starts a transmission by sending two packets to the receiver. When the receiver receives the segments, it returns ACKs (acknowledgements) as confirmation. The sender increments its window by two and sends four packets. This buildup continues with the sender doubling the number of packets it sends until an ACK is not received, indicating that the flow has reached the network's ability to handle traffic or the receivers ability to handle incoming traffic .Slow start does not prevent congestion, it simply prevents a host from causing an immediate congestion state. If the host is sending a large file, it will eventually reach a state where it overloads the network and packets begin to drop. Slow start is critical in avoiding the congestion collapse problem. But new

applications such as voice over IP cannot tolerate the delay caused by slow start and in some cases; slow start is disabled so the user can grab bandwidth. That trend will only lead to problems.

3.2 Fast Transmit and Recovery (RENO)

Fast retransmit and fast recovery are algorithms that are designed to minimize the effect that dropping packets has on network throughput. The fast retransmit mechanism infers information from another TCP mechanism that a receiver uses to signal to the sender that it has received packets out of sequence [10]. The technique is to send several duplicate ACKs to the sender. Fast retransmit takes advantage of this feature by assuming that duplicate ACKs indicate dropped packets. Instead of waiting for an ACK until the timer expires, the source resends packets if three such duplicate ACKs are received. This occurs before the timeout period and thus improves network throughput. For example, if a host receives packet 5 and 7, but not 6, it will send a duplicate ACK for packet 5 when it receives packet 7(but not packet 6).Fast recovery is a mechanism that replaces slow start when fast retransmit is used. Note that while duplicate ACKs indicate that a segment has been lost, it also indicates that packets are still flowing since the source received a packet with a sequence number higher than the missing packet[11].In this case, the assumption is that a single packet has been dropped and that the network is not fully congested. Therefore, the sender does not need to drop fully back to slow start mode but to half the previous rate.

4. Distance Vector Routing Protocol - Congestion Aware Distance Vector (CADV)

In a distance vector routing protocol, every host maintains a routing table containing the distances from itself to possible destinations. A mobile host in an ad hoc network can be viewed as a single server queuing system. The delay of sending a packet is positively correlated with congestion. In CADV [12], each entry is associated with an expected delay, which measures congestion at the next hop. Every host estimates the expected delay based on the mean of delay for all data packets sent in a past short period of time. Currently, the length of the period is equal to the interval between two periodical updates.

When a host broadcasts an update to neighbors, it specifies the delay it may introduce. A routing decision is made based on the distance to the destination as well as the expected delay at the next hop. CADV tries to balance traffic and avoid congestion by giving priority to a route having low expected delay. A CADV routing module consists of three components.

- Traffic Monitor monitors traffic going out through the link layer. Currently it keeps track of the average delay for sending one data packet in recent period of time. The time period is specified by the route maintenance component.
- Traffic control determines which packet is the next to send or drop, and reschedules packets if needed. At present, it supports a drop tail FIFO queue and provides functionality to re-queue packets.
- Route maintenance is the core component. Its functionalities include exchanging information with neighbors, evaluating and maintaining routes, managing the traffic monitor and traffic control components.

CADV outperforms AODV in delivery ratio by about 5%, while introduces less protocol load. CADV introduces higher end-to-end delay than AODV and DSDV do when the number of connections is greater than 10, because it may choose longer route to forward packets. The delay is rather stable with the increase of the number of connections. CADV consumes less power. For the movements of mobile hosts generated by the random waypoint model, the link change and route change are, with a very high probability, linear functions of the maximum speed, and linear functions of the pause time, respectively. The protocol load for the proactive routing protocols (such as DSDV) grows as the number of hosts increases, while that of the on-demand routing protocols (such as AODV) increases with the number of source-destination (S-D) pairs. The proactive approach performs better when the number of S-D pairs is close to the number of hosts. CADV is not congestion adaptive. It offers no remedy when an existing route becomes heavily congested.

5. Congestion Adaptive Routing In AD HOC Networks

CRP[14],[15] protocol tries to prevent congestion from occurring in the first place. CRP uses additional paths (called “bypass”) to reduce packet delay, but tries to minimize bypass to reduce the protocol overhead. Traffic is split over the bypass and the primary route probabilistically and adaptively to network congestion. Hence, 1) power consumption is efficient because traffic load is fairly distributed and 2) congestion is resolved

beforehand and, consequently, CRP enjoys a small packet loss rate. In CRP, every node appearing on a route warns its previous node when prone to be congested. The previous node then uses a "bypass" route bypassing the potential congestion to the first non-congested node on the route. Traffic will be split probabilistically over these two routes, primary and bypass, thus effectively lessening the chance of congestion occurrence. CRP is on-demand and consists of the following components:

(1) Congestion monitoring,

(2) Primary route discovery,

(3)Bypass discovery,

(4) Traffic splitting and congestion adaptivity,

(5) Multi-path minimization, and

(6) Failure recovery.

5.1. Congestion Monitoring

A variety of metrics can be used for a node to monitor congestion status. Chief among these are the percentage of all packets discarded for lack of buffer space, the average queue length, the number of packets timed out and retransmitted, the average packet delay, and the standard deviation of packet delay. In all cases, rising numbers indicate growing congestion. Any of these methods can work with CRP in practice. We further classify the congestion status at a node into 3 levels: "green", "yellow", and "red". A node is said to be "green" if it is far from congested, "yellow" if likely congested, or "red" if most likely or already congested. A bypass is a path from a node to its next green node. The next green node is the first green node at least two hops away downstream on the primary route.

5.2. Primary Route Discovery

To find a route to the receiver, the sender broadcasts a REQ packet toward the receiver. The receiver responds to the first copy of REQ by sending toward the sender a REP packet. The REP will traverse back the path that the REQ previously followed. This path becomes the primary route between the sender and the receiver. Nodes along this route are called primary nodes. To reduce traffic due to route discovery and better deal with congestion in the network, we employ two strategies: (1) the REQ is dropped if arriving at a node already having a route to the destination, and (2) the REQ is dropped if arriving at a node with a "red" congestion status.

5.3. Bypass Discovery

A node periodically broadcasts to neighbors a UDT (update) packet. This packet contains this node's congestion status and a set of tuples {destination R, next green node G, distance to green node m}, each for a destination R that the node has a route to. The purpose is that when a node N receives a UDT packet from its next primary node Nnext regarding destination R, N will be aware of the congestion status of Nnext and learn that the next green node is G which is m hops away on the primary route. If Nnext is yellow or red, a congestion is likely ahead if data packets continue to be forwarded on link N Nnext . Since CRP tries to avoid congestion from occurring in the first place, N starts to discover a bypass route toward node G - the next green node of N known from the UDT packet. This bypass search is similar to primary route search, except that: (1) the bypass request packet's TTL is set to $2 \times m$, and (2) the bypass request is dropped if arriving at a node (neither N nor G) already present on the primary route. Thus, it is not costly to find a bypass and the bypass is disjoint with the primary route, except that they join at the end nodes N and G. It is possible that no bypass is found due to the way the bypass request approaches G. In which case, we continue using the primary route. However, [13] finds that the chance for a "short-cut" to exist from a node to another on a route is significant.

5.4. Traffic Splitting and Congestion Adaptability

At each node that has a bypass, the probability p to forward data on the primary link is initially set to 1 (i.e., no data is sent along the bypass). It is then modified periodically based on the congestion status of the next primary node and the bypass route. The congestion status of the bypass is the accumulative status of every bypass nodes. The key is that we should increase the amount of traffic on the primary link if the primary link leads to a less congested node and reduce otherwise. An example is demonstrated by Figure 1, where the bypass from A is A→ X → Y→ C, from B is B→Y →Z→ E, and from D is D →W→F .

5.5. Multi-path Minimization

To reduce the protocol overhead, CRP tries to minimize using multiple paths. If the probability p to forward data on a primary link approaches 1.0, this means the next primary node is far from congested or the bypass route is highly congested. In this case, the bypass at the current node is removed. Similarly, if the next primary node is very congested (p approaches 0), the primary link is

disconnected and the bypass route becomes primary. To make the protocol more lightweight, CRP does not allow a node to have more than one bypass. The protocol overhead due to using bypass is also reduced partly because of short bypass lengths. Each bypass connects to the first non-congested node after the congestion spot, which should be just a few hops downstream.

5.6. Failure Recovery

A desirable routing protocol should gracefully and quickly resume connectivity after a link breakage. CRP is able to do so by taking advantage of the bypass routes currently available. For instance, in Figure 1, if node C or D fails or moves away, B can take the bypass B →Y→Z →E.

6. ECARP: An Efficient Congestion Adaptive Routing Protocol for Mobile Ad Hoc Networks

The proposed congestion control routing protocol outperform all the other routing protocols during heavy traffic loads. The simulation experiment with five CBR traffic source sessions between to common destination using AODV[17], DSR[16], DSDV and TORA were conducted. The performance metrics are Average Packet Delivery Ratio and Average End-to-End delay. For observation in as constraint situation we have considered only Average Packet Delivery Ratio, In normal case AODV outperforms better than other three routing protocols . The TORA performs better than DSDV. But under constraint situation of same routing protocols behaves differently. With six CBR traffic sources to a common destination, AODV suffers degradation up to 35% whereas DSR suffers only 10% compared to normal situation. TORA suffers degradation of 45% whereas DSDV suffers only 15%. On comparing their performances, it was observed that DSR performs better than other three routing protocols. The main reason for performance degradation in packet delivery ratio is due to packet drops by the routing algorithm after being failed to transfer the data in the active routes. There are several reasons for packet drops such as network partitioning, link break, collision and congestion in the ad hoc networks. The main important property of routing algorithm is quick link recovery through efficient route maintenance. Therefore, the DSR routing protocol has fast reaction for link recovery and finds alternative path (during congestion) in compared with AODV and other routing protocols in the given situation. AODV keeps only the active and removes the state ones. Therefore, unavailability of the alternate routes leads to route discovery by the source node. The congestion will be high when multiple CBR sources send data to a single destination. In AODV, the intermediate, nodes are unable to send the data packets, link break situation perceived by AODV sends route error or finding new route through source will result in packet drops resulting in degradation of packet delivery ratio, increase in Average End-to-End delay and increase in Routing overhead. Thus AODV ensures the high availability of alternative routes and reduce the rate of broken route removal process.

7. ECARP Congestion Control Algorithm

This algorithm provides solution to improve routing protocols due to constrained environment.

Step1: check the occupancy of link layer buffer of node periodically, Let C be the congestion status estimated.

Step2: Compute Cs = Number of packet buffered in Buffer Buffer size

Step 3: Set the status for Congestion. It can be indicated by three statues “Go”, “Careful” and “Stop” . [“Go” indicates there is no congestion with Cs ½ “careful” indicates the status likely to be congested with ½ Cs ¾ and “Stop” indicates the status likely to be congested ¾ Cs 1.]

Step4: Invoke congestion control routine when link failed event has occurred in data transfer with using active route or ¾ Cs 1.

Step 5: Assume that neighbor will have alternate route or noncongested route to the destination.

Step 6: Make Query to non-congested neighbors for route to destination.

Step 7: After obtaining the routes from the neighbors, select route with minimum hops.

Step 8: Once route is finalized start sending the data packets through non-congested route.

Step 9: If there is no alternative route to destination then start splitting the traffic to the less congested route.

Step 10: Traffic splitting effectively reduces the congestion status at the next main node

8. Conclusions

As the congestion control is the most important factor of any packet switching network, the whole performance and accuracy of network is directly related to it, the congestion control becomes more important. We briefly survey of various congestion control algorithms. It shows that at present there is no single algorithm that can resolve every problems of congestion control on computer networks. Further research work is needed in this direction.

References

[1] T. Azuma, and M. Fujita :``Congestion Control in Computer Networks" , Journal of The Society of Instrument and Control Engineers, Vol. 41, No. 7, pp. 496--501 (2002--7)

[2] D. Cavendish, M. Gerla, and S. Mascolo, “A control theoretical approach to congestion control in packet networks,” IEEE/ACM Trans .Network., vol. 12, no. 5, pp. 893–906, Oct. 2004.

[3] C.-Y. Wan, S. B. Eisenman, and A. T. Campbell, “CODA: Congestion Detection and Avoidance in Sensor Networks,” in Proc. ACM SenSys, Nov.2003.

[4] Partho P.Mishra, Hemant Kanakia, “A Hop-by-hop Ratebased Congestion Control Scheme,” In Proc. ACM SIGCOMM’92: 112-123.

[5] B. Hull, K. Jamieson, and H. Balakrishnan, “Mitigating Congestion in Wireless Sensor Networks,” in Proc. ACM Sensys, Nov. 2004.

[6] Sankarasubramaniam, Y., Ozgur, A., Akyildiz, I.: ESRT Event-to-Sink Reliable Transport in Wireless Sensor Networks. In: the Proceedings of ACM Mobihoc, pp. 177–189. ACM Press, New York (2003)

[7] C.T. Ee and R. Bajcsy, “Congestion Control and Fairness for Many-to-one Routing in Sensor Networks,” in Proc. ACM Sensys, Nov. 2004.

[8] Priority-based Congestion Control in Wireless Sensor Networks Chonggang Wang1, Kazem Sohraby1, Victor Lawrence2, Bo Li3, Yueming Hu4Proceedings of the IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC’06)

[9] Md. Obaidur Rahman, Muhammad Mostafa Monowar and Choong Seon Hong, “A QoS Adaptive Congestion Control in Wireless Sensor Network” in IITA, Nov. 2006.

[10] Van Jacobson. Modified TCP Congestion Control Avoidance Algorithm. end-2-end-interest mailing list, April 30, 1990

[11] W. Stevens. TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms, January 1997, RFC 2001.

[12] Xiaoqin Chen, Haley M. Jones and Jayalath, "Congestion-Aware Routing Protocol for Mobile Ad Hoc Networks" IEEE 66th conference on Vehicle Technology, pp.21-25, October 2005.

[13] S. Ramanathan, M. E. Steenstrup, “A survey of routing techniques for mobile communications networks, mobile networks and applications,” Vol. 1, pp. 98–104, 1996

[14] H. Raghavendra and D.A. Tran, “Congestion Adaptive Routing in Ad Hoc Networks (Short Version),” Proc. ACM Int’l Conf. Mobile Computing and Networking (MOBICOM), Oct. 2004.

[15] H. Raghavendra and D.A. Tran, “Congestion Adaptive Routing in Ad Hoc Networks” IEEE Transactions on Parallel and Distributed Systems, Vol. 17, No. 11 , November 2006.

[16] J. Broch, D. Johnson, and D. Maltz, “The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks,” IETF Internet draft, Oct.1999.

[17] C.E. Perkins, E.M. Belding-Royer, and I. Chakeres, “Ad Hoc On Demand Distance Vector (AODV) Routing,” IETF Internet draft, Oct. 2003.