Background Data Transfers with Minimal Delay Impact Using Congestion Control in WSN

[1]Lohith N, [2]Chinnaswamy C.N, [3]T.H Sreenivas

[1]PG Student, [2,3]Associate Professor
[1,2]Department of ISE
[3]Department of CSE
[1,2]The National Institute of Engineering, Mysuru, India
[3]VVCE, Mysuru

Abstract- Congestion control protocols are used to avoid the congestion in the network, the congestion occurs due to the more data transfer in the network ,In existing system to overcome this congestion problem they proposed distributed congestion protocol for heterogeneous traffic using CSMA . Another protocol TACCP is used to avoid packet loss caused by traffic congestion. Similarly many congestion protocols are used. In the presence of even a very long TCP flows, this behavior can cause bandwidth starvation, this effect on the download delays of delay-sensitive TCP flows. In this paper states about the fundamental problems of designing congestion control protocols for background traffic with the minimum impact on short TCP flows while achieving a certain desired average throughput over time. The corresponding optimal policy under various assumptions on the available information is obtained. We give tight bounds of the distance between TCP-based background transfer protocols and the optimal policy, and Identify the range of system parameters for which more sophisticated congestion control makes a noticeable difference

Index Terms - wireless sensor networks; congestion control protocols;bandwidth.

I. INTRODUCTION

The success of the internet architecture is the ability to replace currents and future needs of very diverse applications. While transferring the large size of data from the source node the sink node does not have the ability to receive the data. To overcome this problem transport protocol is used namely Transmission Control Protocol (TCP).This protocol assigns the bandwidth to every node fairly. This is useful for congestion control algorithm which intends to ensure equal sharing between flows.

But there are problems when all internet flows use the same protocol as applications do not equally value download delay. Web browsing and media streaming are delay-sensitive tasks where low web-page download delay and low initial playback latency is desirable, respectively. On the other hand, background data transfers such as large batch software or database updates are in different to small temporal variations of their bandwidth share, provided the data volume downloaded over a long time period is sufficient. It is well known from scheduling theory that shorter jobs or jobs with tighter deadlines should be assigned higher priority. Hence using TCP as the common transport protocol creates unnecessary delays to short and delay-sensitive flows.

A possible solution, violating the end-to-end principle of the internet architecture is for the internet service providers (ISPs) to intervene and throttle the bandwidth assigned to background data leaving more space for delay-sensitive traffic or offering some form of prioritization. But this is not in many cases an efficient solution, since the ISPs cannot have the necessary information on how much throttling is necessary, and for which flows [1].Also unjustified throttling of traffic can have serious side effects for the ISP business, e.g., legal actions taken by disaffected end users [2].

The internet engineers have developed end-to-end congestion control protocols for background data transfers called as 'less-than-best-effort' (LBE), like TCP-LP [3], TCP-nice [4], uTorrent transport protocol [5], LEDBAT [6]. These are typically designed to emulate a low priority transport class which yields to TCP traffic, but this behaviour can have a serious drawback under the presence of 'long' TCP flows, i.e., persistent or extremely long-lasting and always active flows. In principle, during the time in which long TCP flows compete with ideal low priority flows, the latter suffer from bandwidth starvation and so their number grows arbitrarily as new low priority flows continue to arrive.

II. RELATED WORK

The subject of fairness between different internet flows is intrinsically linked to congestion control and has been studied extensively over the past decades under different perspectives, e.g., see [10] and references there in. In [7], [8], and [9] the effect of congestion control on the number of ongoing file transfers and download delays is studied .This paper take a similar viewpoint by considering a model where flow-level dynamics are described by a Markovian process, and ignore congestion window dynamics and packet level effects. Deb et al. [10] consider a flow-level model of a large system with many long and short flows. They consider the optimization of congestion controllers of all flows -background, long and short- by maximizing a social welfare function which includes the average utility obtained by background traffic and the delay caused to short flows. Since we assume that part of the traffic, namely long and short, uses TCP for its transport and cannot be optimized, the optimal policies differ considerably from the ones in [10].

In this paper its unnecessary details and assumed to be comprised by

- TCP traffic - which we do not control: it consists of

(i) The long TCP flows
(ii) A stream of TCP flow arrivals

Referred to as ‘short’, with each being a few orders of magnitude shorter2 than long flows, and which consume an average fraction ρ of link capacity in 2. For example, if long flow sizes are of the order of GB, the sizes of short flow are in the MB range.

- Traffic that we control: these are the ‘Controlled Background Flows (CBFs)’, i.e., flows carrying background data whose congestion control protocol we optimize. Each CBF originates at some edge of the network and models either the transfer of a single file of a very large size compared to the sizes of short flows, or a stream of file-transfer requests with sizes comparable to short flows. A natural application of this model is to systems where a level of aggregation is possible, such as file-sharing peers or content servers.

III. BANDWIDTH SHARING FOR BACKGROUND FLOWS

There are different types of model are discussed as follows:

- Basic Model
- Optimal Sharing Under Full Information
- Optimal Sharing Within a Class of Policies
- A weighted TCP Sharing Policy (wTCP)

A. Basic Model

In this basic model they consider a link of capacity C shared by a set of CBFs, long TCP flows, and a dynamically arriving stream of short TCP flows. The latter concern transfers of files with independent and exponentially distributed file sizes, of mean $\mu-1$, and arrive at the link according to a Poisson process with rate λ arrivals per unit time. Because the size of long TCP flows is assumed to be much greater than $\mu-1$, the rate at which flows of this size arrive in the system is orders of magnitude less than λ. This implies that the timescales in which the number of short and long flows vary are widely disparate.

B. Optimal Sharing Under Full Information

When the event occurs at large data flows, at the sink node when hotspot occurs the packets will be dropped near the sink node. This process is called as sink congestion. The main reason to occur sink congestion is battery power of the nodes that are near the sink exhausted quickly. One of the solutions to overcome this problem is to place the multiple sink nodes at a specific distance region.

C. Optimal Sharing Within a Class of Policies

Translating a threshold policy into an end-to-end congestion control algorithm is a challenging task because the number n of ongoing short TCP flows is not directly observable, and so it must be inferred through some indirect way. The natural way to do this is through some end-to-end observable measure of congestion such as packet loss and/or delay, which varies monotonically with n.

D. A weighted TCP Sharing Policy (wTCP)

In this section we consider an implementable policy which is easier to implement and can be thought of as a weighted variant of TCP; thus we call it weighted TCP (wTCP). It is appealing because the delay is not much larger than the optimal under full information, when ρ is close to 1.

IV. IMPLEMENTATION PHASE

- Network configuration and deployment
- TCP Routing Phase
- TCP Based Congestion controller
- Optimal Sharing by congestion feedback
- Dynamic Arriving Back group flows
- Performance analysis

1. Network configuration and deployment

To begin with characterize the Network setup parameters i.e., indicate the quantity of hubs ,beginning vitality, MAC, engendering, Receiver control, rest control, transmission control, Channel Type, Propagation or TwoRayGround i.e., radio-proliferation show, arrange interface (Phy/Wireless Phy), MAC type(Mac/802_11),interface line type(CMUPriQueue), connect layer sort, reception apparatus demonstrate (Antenna/OmniAntenna), maxpacket in ifq, number of portable hubs, X pivot separate, Y hub remove Initial Energy, Initial vitality in Joules. At that point convey every one of the hubs into the system with some moving speed. The system stack for a portable hub comprises of a connection layer (LL), an ARP module associated with LL, an interface need line (IFq), a macintosh layer (MAC), a system interface (netIF), all associated with the channel. These system segments are made and plumbed together in OTcl. The pertinent Mobile Node technique include interface ().

- Create the occurrence for the super class Simulator and make utilization of this reference variable for making and indicating the parameters for the hub.
- Create the nam petition for summoning the nam window with the set charge and opening the nam record in the compose mode .For this document reference variable give the order ns-nam trace-all.
- Creating the topology with set topo charge and indicating the kind of the topology as flat grid and determining x value and system.
- Configuring the hubs by determining the estimations of the system parameters.
- Creating the hubs utilizing the for circle and "$ns-hub" order.
- Assign the positions for every one of the hubs with the setdest order and x value, y value
- Attach the u dp specialist to the hub.
- Attach the CBR activity from source to sink by setting the bundle estimate, parcel interim.
- Connect the specialists

Link Layer: The main distinction being the connection layer for portable hub, has an ARP module associated with it which settle all IP to equipment (Mac) address transformations. Ordinarily for all friendly (into the channel) bundles, the parcels are passed on to the LL by the Routing Agent. The LL pass on bundles to the interface line. For every single approaching bundle, the mac into sh layer hands up parcels to the LL which is then given off at the node-entry-point

ARP: The Address Resolution Protocol (executed in BSD style) module gets inquiries from Link layer. On the off chance that ARP has the equipment address for goal, it composes it into the mac into sh header of the parcel. Else it communicates an ARP question, and stores the bundle incidentally. For every obscure goal equipment address, there is a support for a solitary parcel. In case extra bundles to a similar goal is sent to ARP, the prior cushioned parcel is dropped. Once 151 the equipment address of a bundle's next jump is known, the parcel is embedded into the interface .

Interface Queue: The class PriQueue is executed as a need line which offers need to directing convention bundles, embeddings them at the leader of the line. It underpins running a channel over all parcels in the line and evacuates those with a predefined goal address.

Mac Layer: ns-2 has utilized the usage of IEEE 802.11 conveyed coordination work (DCF) from CMU. Beginning with ns-2.33, a few 802.11 executions are accessible.

Tap Agents: Operators that subclass themselves as class Tap characterized in mach can enrol themselves with the mac into sh question utilizing technique install Tap (). On the off chance that the specific Mac convention grants it, the tap will indiscriminately be given all parcels gotten by the mac into sh layer, before address sifting is finished.

Network Interfaces: The Network Inter phase layer fills in as an equipment interface which is utilized by versatile hub to get to the channel. The remote shared media interface is executed as class Phy/WirelessPhy. This interface subject to crashes and the radio spread model gets bundles transmitted by other hub interfaces to the channel. The interface stamps each transmitted parcel with the meta-information identified with the transmitting interface like the transmission control, wavelength and so forth. This meta-information in packet header is utilized by the proliferation demonstrate in getting system interface to decide whether the bundle has least energy to be gotten or potentially caught or potentially distinguished (transporter sense) by the accepting hub. The model approximates the DSSS radio interface.

Radio Propagation Model It utilizes Friss-space weakening (1/r2) at close separations and a guess to two beam Ground (1/r4) at far separations. The estimate accepts specular reflection off a level ground plane.See ~ns/tworayground.{cc,h} for execution. Recieving wire An Omni-directional reception apparatus having solidarity pick up is utilized by versatile hubs.

2. TCP Routing Phase

This is achieved using only a handful of transport protocols, mainly Transmission Control Protocol (TCP) and its variants, which in essence allocate network bandwidth to flows continuously so as to achieve fair sharing at all times. Indeed TCP ‘fairness’ or ‘friendliness’ has become a common prescription for congestion control algorithms which intends to ensure equal sharing between flows. Hence using TCP as the common transport protocol creates unnecessary delays to short and delay-sensitive flows.

3. TCP Based Congestion controller

When a network is very long in nature then there is a need to compress it to extend the lifetime and used WSN in energy efficient way. One approach is to used distributed compressive sampling. The nodes in the network decides whether to use compression and forwarding to the next node. So, that the number of packets to be transmitted will be reduced and only important part of the data will be sent to the destination.

4. Optimal Sharing by congestion feedback

It solves the same optimization problem but restricted to a set of policies that use only information available at the edges of the network by reacting to congestion. It has a simple form if link congestion is above some level, the controller of the CBF traffic sends no data; else it sends at a high enough rates to keep congestion at this constant level. This policy performs asymptotically as the optimal full information policy for $\rho \to 1$ (Theorem 4), and numerical analysis shows that it is within few percents of the latter even for smaller values of ρ.
If there is a explicit target for the average throughput, our results suggest a simple adaptive algorithm: use the optimal implementable congestion controller while slowly adapt the congestion threshold of the algorithm to achieve the desired throughput.

5. Dynamic Arriving Back group flows

In the previous sections we used a system model with a fixed number of background flows. This is justified when there is an infinite amount of background data readily available for transfer. In this section we consider a link model as the one in A but where the CBF is comprised by a stream of ‘micro-flows’ of finite duration, arriving according to a Poisson process with rate λb. Each micro-flow is associated with the download of a file with an exponentially distributed size with mean $1/\mu b$. Also the file sizes are assumed to be independent across different flows. If the CBF offered rate $\lambda b/\mu b$ is less than the excess capacity and no amount of flow is lost, the fraction of excess capacity consumed by the CBF (or equivalently by its micro-flows) is $f = \lambda b/[\mu bC(1 - \rho)]$. We also define the load of the CBF to be $\rho b = \lambda b\ C\mu b$.

6. Performance analysis

In this mathematical operations are performed based on the above all the operations then results will be stored into the x graphs.

V. CONCLUSION

Our results suggest that in many cases the performance of high-performance tools are degraded across shared networks. We conclude that the fundamental problem of designing congestion control protocols for background traffic with the minimum impact on short TCP flows while achieving a certain desired average throughput over time. The corresponding optimal policy under various assumptions on the available information is obtained

REFERENCES

[1] B. Briscoe, “A fairer, faster Internet,” IEEE Spectr., vol. 45, no. 12, pp. 38–43, Dec. 2008.

[2] N. Anderson. (Jul. 2010). Claim Your $16! Comcast P2P Settlement Now Final. [Online]. Available: http://arstechnica.com/techpolicy/2010/07/claim-your-16-comcast-p2p-settlement-now-final/

[3] A. Kuzmanovic and E. W. Knightly, “TCP-LP: Low-priority service via end-point congestion control,” IEEE/ACM Trans. Netw., vol. 14, no. 4, pp. 739–752, Aug. 2006.

[4] R. Venkataramani, R. Kokku, and M. Dahlin, “TCP Nice: A mechanism for background transfers,” in Proc. 5th Symp. Oper. Syst. Design Implement. (OSDI), Boston, MA, USA, Dec. 2002, pp. 329–344.

[5] A. Norberg. uTorrent Transport Protocol,accessed on Mar. 15, 2016. [Online]. Available: http://www.bittorrent.org/beps/bep_0029.html.

[6] S. Shalunov, G. Hazel, J. Iyengar, and M. Kuehlewind, Low Extra Delay Background Transport (LEDBAT), document IETF RFC 6817, 2012.

[7] R. Annie Uthra, S. V. Kasmir Raja, A. Jeyasekar, Anthony J. Lattanze, "A probabilistic approach for predictive congestion control in wireless sensor networks", Zhejiang University-science (Computers & Electronics), vol. 15, no. 3, pp. 187-199, 2014.

[8] Zhi Liu , Yun Zhang, C.L. Philip Chen, "Adaptive mechanism-based congestion control for networked systems", Systems Science, vol. 44,no.3,pp. 533-544, 2013.

[9] Akbar Majidi, Hamid Mirvaziri, "BDCC: Backpressure routing and dynamic prioritization for congestion control in WMSNs",Computer Network and Information Security, vol. 5, pp. 29-34, 2014.

[10] Charalambos Sergiou, Vasos Vassiliou, Aristodemos Paphitis,"Congestion Control in Wireless Sensor Networks Through Dynamic.