AVOCA – A Vehicle Oriented Congestion Control Algorithm

Yi Huang, Enda Fallon, Yuansong Qiao, Marcus Rahilly, Brian Lee

Software Research Institute,

Athlone Institute of Technology,

Athlone, Ireland.

email : A00168956@student.ait.ie, efallon@ait.ie, ysqiao@ait.ie ,mrahilly@ait.ie, blee@ait.ie

***Abstract*—IP transport layer protocols such as TCP and SCTP have inherited legacy features which are fixed line in nature. One of the major responsibilities of a connection oriented transport layer protocol is congestion control. We illustrate that for a mobile vehicular session, the aggressive reaction of standard congestion control procedures to packet loss results in communication blocking. We suggest that for vehicle based systems, such as underground trains, coverage holes are an element of a normal operating environment. We propose A Vehicle Oriented Congestion control Algorithm (AVOCA) which considers network coverage holes when implementing congestion control. We illustrate that the standard congestion control procedures in TCP and SCTP can result in communication delays of up to the configured RTO.max value. We illustrate that AVOCA has significant throughput improvement while observing network fairness policy.**

***Keywords* – Vehicular Systems, Congestion Control, SCTP, TCP**

I INTRODUCTION

Many IP protocols have evolved from fixed line environments. Even recent protocols such as SCTP have inherited legacy features, such as congestion control, which are fixed line in nature. In this paper we illustrate that the congestion control procedures implemented in connection oriented transport layer protocols such as TCP and SCTP are unsuitable for modern vehicle systems in which network connectivity is intermittent in nature. As connection oriented protocols TCP and SCTP must guarantee in order delivery of packets to the receiving application layer. We illustrate that the implementation of RTO binary exponential back off delays the detection of packet loss, causing a new form of congestion window blocking communication failure.

We focus on the optimisation of network throughput for set route vehicles such as public transport busses and trains. Such vehicles typically operate in preconfigured routes which are repeated at routine intervals sometimes many times a day. We illustrate that in such an environment existing transport layer mechanisms fail to fully utilise intermittent connectivity as the vehicle enters zones of coverage. We illustrate that this performance degradation results from the legacy fixed network oriented congestion control and RTO calculation mechanisms. We illustrate that in a worst case scenario communication will be delayed by up to the value specified by RTO.max. By default RTO.MAX is set to 60 seconds in TCP and SCTP.

In order to alleviate this congestion window blocking performance degradation we propose AVOCA – A Vehicle Oriented Congestion control Algorithm. AVOCA is a cross layer congestion control algorithm which utilises a layer 2 performance threshold to control transport layer packet transmission. Unlike traditional congestion control mechanisms AVOCA assumes that network connectivity will be intermittent. When the Layer 2 performance threshold is exceeded, indicating that the vehicle is entering a zone of wireless coverage, congestion control parameters are reset and packet transmission is initiated. When the vehicle moves from the zone of coverage the performance and RSS is less than the threshold; packet transmission will be terminated and all congestion control parameters will be frozen. In a partially reliable configuration the final packet transmitted will be a Forward Transmission Sequence Number (TSN) indicating that all in flight packets should be ignored.

We evaluate AVOCA using a simulated scenario recreating the Circle line on the London underground system. The Circle line was selected as it contained stations with varying levels of passenger usage. Results presented illustrates that the AVOCA approach removes the new form of congestion window blocking thereby enabling the transport layer protocols to transmit during intermittent network connectivity. Such an approach significantly improves overall throughput while maintaining the principles of Internet fair usage.

This paper is organised as follows; related work is described in section 2. A technology overview is provided in 3. In section 4 AVOCA is described. In section 5 the usage scenario is described. A simulated evaluation is described in section 6. Results are presented in section7. A detailed illustration of the TCP congestion window blocking is provided in section 8. Conclusions are presented in section 9.

II RELATED WORK

A number studies have investigated how network performance can be optimised by predicting network holes [1][2]. These studies are ad hoc network based and assume an autonomic approach which allows the network to self learn/configure. As an end point oriented solution our approach has no ability to change network configuration.

Other solutions propose to optimise the calculation of Retransmission Time Out (RTO) calculation in order to optimise throughput. In [3] explicit packet drop reports are generated. A new chunk type, Packet Drop Chunk (PKTDROP), is introduced which is generated by middleware boxes and returned to the endpoints. It is intended that such an approach will enable endpoints to differentiate packet loss due to path degradation from packet loss due to Internet congestion. Such an approach however assumes that some form of network communication is available. In our scenario vehicle movement has resulted in catastrophic network failure.

When packet loss occurs binary exponential back-off is implemented. The repeated implementation of binary exponential back-off can significantly reduce throughput. In [4] a Fixed-RTO algorithm for TCP is proposed which does not implement binary exponential back-off for every packet loss, rather it implements binary exponential back-off repeatedly up to a threshold value, subsequent packet loss does not result in binary exponential back-off. In [5] this mechanism is shown to result in performance improvement for in ad hoc WLAN networks. As the study is ad hoc rather than infrastructure mode based, it assumes that packet loss is not as a result of Internet congestion. The selective implementation of binary exponential back-off in [5] is coarse grained as it is based on a threshold value, below the threshold implement binary exponential back-off, above it do not implement binary exponential back-off. The mechanism we propose in [6] is targeted towards infrastructure based configurations. It utilises separate RTO components for the WLAN access and Internet components of RTO and is therefore more reflective of actual network performance.

The provision of seamless communications for vehicular systems is an increasingly active area of research. In [7] the requirements for effective communication for such systems are discussed. As well as congestion control, the authors suggest that issues such as vehicle speed and application Quality of Service (QoS) requirements need to be considered. In [8] two traffic rate control schemes to address the congestion control problem for short range communication networks are proposed. Channel occupancy time measured at MAC layer is used to detect channel congestion status. A congestion signal is then forwarded to the application layer for adaptive rate control. Such an approach is application specific; we advocate a transport layer oriented approach to congestion control which provides support to all applications. [9] proposes a context aware cooperative congestion control policy. The policy exploits the traffic context information of each vehicle to in order to reduce channel load and reduce unnecessary interference and decrease channel load. Such an approach is useful when optimising existing transmission capacity. It does not address the issue of congestion control behaviour caused by network holes.

III TECHNOLOGY OVERVIEW

a) Connection Oriented Congestion Control

Connection oriented protocols such as TCP and SCTP use the following three variables to control congestion; Receiver advertised window size (rwnd), Congestion control window (cwnd) and Slow-start threshold (ssthresh). The rwnd is set by the receiver for the entire association based on its available buffer space. The cwnd controls the number of unacknowledged packets that may be in transit end-to-end and is set per path. There are 2 phases defined for congestion control in SCTP; slow start and congestion avoidance. The ssthresh is used to distinguish between the 2 phases. During the slow start phase the cwnd grows exponentially. Every time an acknowledgement is received during the slow start phase the cwnd grows by the number of packets acknowledged. This continues until the ssthresh is reached or a packet loss occurs. When the slow start phase is complete SCTP enters the congestion avoidance phase. During congestion avoidance the cwnd grows in a linear manner at a rate of 1 packet per Round Tip Time (RTT). SCTP slow-start is used to probe the network to determine the available capacity at start-up or after repairing loss detected by the retransmission timer. During slow start the cwnd is increased each time an acknowledgment is received. It increases the cwnd by number of packets acknowledged.

In the congestion avoidance phase SCTP uses the additive increase/multiplicative-decrease (AIMD) algorithm. AIMD combines linear growth of the cwnd with an exponential reduction when congestion occurs. The cwnd is increased by one packet every RTT until a loss is detected. When a loss is detected, the policy changes to multiplicative decrease. When packet loss is detected from a Selective Acknowledgement (SACK) an endpoint should half the cwnd and set the ssthresh to this value also. When the retransmission timer expires on a path, SCTP should perform slow-start by setting ssthresh = cwnd/2 and cwnd= 1. This will ensure only 1 packet of data is sent and acknowledged before another packet is sent

When an SCTP peer receives a packet out of sequence, it sends a SACK for the out of sequence packer and includes a gap block for missing TSN's. The fast retransmit algorithm considers 3 SACK's to mean that the packet is lost and it will retransmit the packet without waiting for a RTO timeout. The fast retransmission algorithm continues to control the transmission of new packets until all lost packets have been retransmitted.

b) Calculation of RTO

For TCP the RTO algorithm is defined in [10]. At start-up RTO is set according to to a starting value RTO.Initial = 3000.

$$RTO = RTO.Initial$$

The Round-Trip Time (RTT) of a path is the basis of RTO calculation. RTT measurements are not taken for retransmitted packets following Karn's algorithm [11]. According to [12] RTT measurement for a path should be made once for every round trip. RTO is calculated according to Jacobson's algorithm [13] as follows:

$$RTO = SRTT + 4xRTTVAR$$

where RTTVAR is the mean deviation of the RTT samples. Once the first RTT measurement is taken the Smoothed RTT (SRTT) and RTT Variance (RTTVAR) are initialised to:

$$SRTT = RTT.1^{st}$$

$$RTTVAR = {}^{RTT.1st}\!/_{2}$$

Each time SCTP gets a new RTT measurement RTT.new, SRTT and RTTVAR will be updated as follows:

$$RTTVAR.new = (1-\beta) \times RTTVAR.old + \beta \times (SRTT.old - RTT.new)$$

$$SRTT.new = (1-\alpha) \times SRTT.old + \alpha \times RTT.new$$

where the values α=.125 and β =.25 are recommended by [13]. The new RTO is then:

$$RTO = SRTT.new + 4 \times RTTVAR.new$$

In [14] it was suggested that a conservative minimum RTO of 1000ms be used to avoid spurious retransmissions. If the new RTO is less than RTO.Min, it will be set to RTO.Min. If the new RTO is greater than RTO.Max (60s), it will be set to RTO.Max. In response to Internet congestion collapse of the mid 1980s it was suggested [13] that a back off mechanism be employed when congestion was detected. The suggested alterations were formally adopted [10] in 1989. Every time a transmission timeout occurs for an address, the RTO for this address will be doubled:

$$RTO = RTO\ x2$$

IV AVOCA ALGORITHM

The traditional fixed line origins of connection oriented transport layer protocols makes then more suitable to scenarios in which there exists a permanent underlying physical network for the duration of the communication session. Seamless network mobility has attempted to preserve the illusion of continuous network connectivity by abstracting the transition from network to network. Such an approach however assumes the existence of some form of underlying network. For modern vehicular systems "network holes" are common. Below we outline the structure of the AVOCA algorithm which enables effective communication in such an environment.

```
Struct AVOCAData
  param Boolean TransmitMode=false;
  param Boolean TransmitStarted=false;
  param Boolean L2Configured=false;
  param Boolean L3Configured=false;
  param int RSSThres;

Routine::AVOCA()
   for(;;){
     if(TransmitMode==true){
       if(TransmitStarted==true){
         CongestionCtrl();
       }
       else{
          initCongestionParam();
          CongestionCtrl();
       }
     }
     else{
       if(L2Configured=true){
         if(L3Configured=true){
           int currentRSS = readRSS();
           if(currentRSS>RSSThres){
              TransmitMode=true;
           }
         }
         else{
             initiateMobileIP();
         }
       }
       else{
         wirelessProbing();
       }
     }
  }
```

V EVALUATION ENVIRONMENT

In order to evaluate the AVOCA algorithm we create a scenario using the London underground system. In particular we evaluate a 107 minute round trip on the Circle line. Figure 1 illustrates the stations on this line while Table 1 details the relative geographical position and the intermediate travel time.

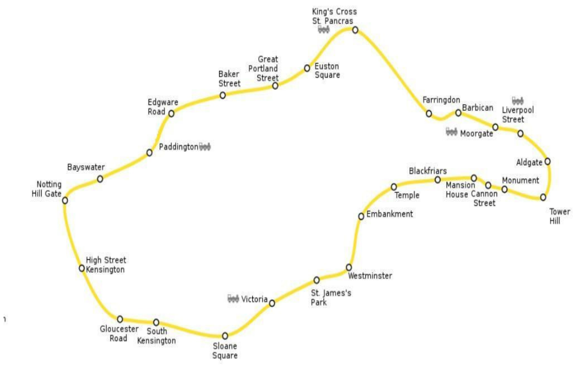

Figure 1. Circle Line Stations Geographical Map

The total length of a cycle journey is 30.21km. The average speed of a train is 34.2km/h. In our simulation we assume that a train halts at each station for a standard period of 2 minutes.

Station	Distance From Previous	Travel Time(sec)	Arriving	Departing
Baker Street	1164	123	123	243
Great Portland Street	1505	159	402	522
Euston Square	932	99	621	741
King's Cross St Pancras	1906	202	943	1063
Farringdon	2119	224	1287	1407
Barbican	838	89	1496	1616
:	:	:	:	:
Westminster	691	73	3808	3928
St. James's Park	1046	111	4039	4159
Victoria	1113	118	4277	4397
Sloane Square	1487	157	4555	4675
South Kensington	1947	206	4881	5001
Gloucester Road	1008	107	5107	5227
High Street Kensington	1276	135	5363	5483
Notting Hill Gate	1041	110	5593	5713
Bayswater	1023	108	5821	5941
Paddington	1815	192	6133	6253
Edgware Road	616	65	6318	6438

Table 1: Relative Distance between Stations

VI SIMULATION CONFIGURATION

In order to evaluate the performance of our AVOCA algorithm an NS2 [15] simulation was created. The simulation used the geographical information from the previous section to define the location of wireless signal coverage on the circular route. Figure 2 illustrates the simulation topology corresponding to Table 1. The characteristics of the wireless channel within each station were configured using data recorded experimentally using the Netstumbler and IXChariot network analysis packages. Using this approach the delay and loss experienced by a mobile client located on the train as it entered and departed a station could be determined. Using data from [16] the station passenger usage was estimated. The busiest stations were those co located with train stations; Paddington, Kings Cross, Victoria. The stations with the lowest passenger numbers were Barbican, Moor Gate and Mansion House. Using this passenger number information we categorized the stations representing high, moderate high, moderate low and low passenger usage. Each of the 27 stations on the line was allocated one of these ratings to represent the load on the installed wireless network.

Figure 2 illustrates the simulation configuration. A mobile client on the train downloads data such as advertising from a content server using FTP. As the train enters stations wireless connectivity is established and data transmission can continue. As the train departs the station wireless coverage is terminated, when the train enters the following station the previous session must be reinitiated.

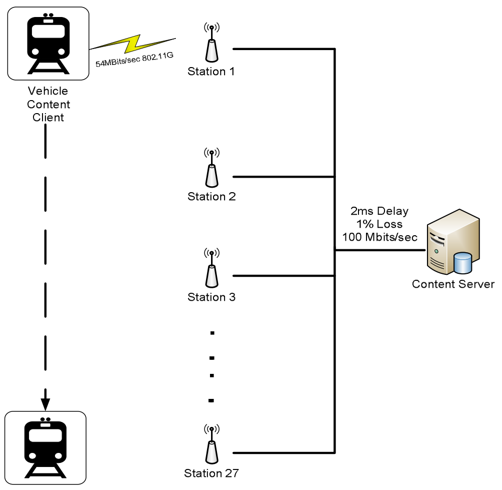

Figure 2. Simulation Configuration

Our analysis involved the simulation of single cyclical journeys starting and ending at Paddington. In order to evaluate how time of day affected performance 3 usage scenarios were evaluated; normal operation, Rush hour and bank holiday. Our results were compared against the TCP approach defined in [17].

VII Results

In this section we evaluate our results for circular routes of 107 minutes in duration starting and ending at Paddington station. Figure 3 compares the performance of our AVOCA approach against TCP for the usage scenarios; normal operation, rush hour and bank holiday. It illustrates the accumulated data transmitted during each cycle.

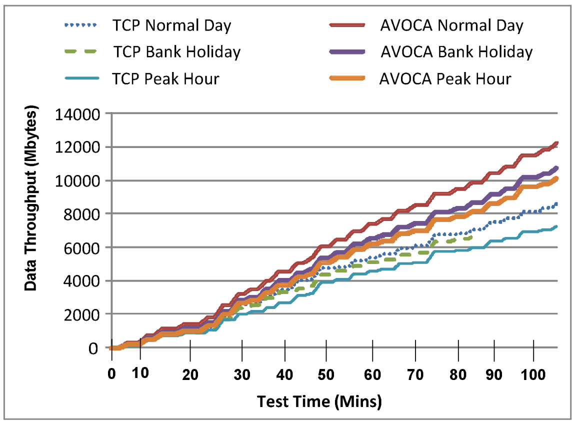

Figure 3. Accumulated Data Transmitted AVOCA/TCP

Figure 3 illustrates that our AVOCA algorithm significantly out performs the standard TCP approach. Table 2 indicates the final total data transmitted by the TCP and AVOCA algorithms for each of the usage scenarios. It illustrates that for normal day operation the AVOCA algorithm has a 42% performance improvement. For bank holiday and rush hour usage AVOCA has a 38% and 40% performance improvement respectively. .

	Normal Day	**Bank Holiday**	**Rush Hour**
TCP	861.44MB	780.36MB	724.19MB
AVOCA	1221.39MB	1073.11MB	1010.76MB

Table 2: Performance Comparison AVOCA/TCP

Table 3 illustrates the throughput experienced at each station based on passenger usage and the time of day.

Usage Type	**Algorithm**	**Station Congestion**	**Throughput (Mbytes)**
Normal Day	**TCP**	Low	177
		Moderate Low	128
		Moderate High	93
		High	59
	AVOCA	Low	230
		Moderate Low	177
		Moderate High	138
		High	100
		Moderate High	117
		High	75
Bank Holiday	**TCP**	Low	175
		Moderate Low	122
		Moderate High	75
		High	47
	AVOCA	Low	224
		Moderate Low	162
		Moderate High	117
		High	75
Rush Hour	**TCP**	Low	173
		Moderate Low	123
		Moderate High	65
		High	42
	AVOCA	Low	224
		Moderate Low	158
		Moderate High	109
		High	59

Table 3: Performance Comparison by station Type

VIII Analysing the AVOCA Performance Improvement

In order to explain the performance improvement of the AVOCA algorithm in comparison to the standard TCP we evaluate a sample section, Paddington to Edgware Road, in more detail. Edgware Road is 616M from Paddington. Travelling at an average speed of 12m/sec requires 50 seconds journey time. In order to illustrate the performance differential between AVOCA and TCP we configure a generic loss rate and delay of 5% and 16ms respectively in both stations. While the train is between the stations we configure a 100% loss rate. We record throughput when the train enters Paddington. Both approaches have similar performance while utilising the wireless coverage in Paddington. At 120 seconds the train leaves Paddington. At 170 seconds the train enters Edgware Road. When Mobile IP address configuration has been complete, the AVOCA algorithm reinitialises its congestion control parameters and begins packet transmission. The TCP approach however delays packet transmission until 182 seconds, a delay of 12 seconds following layer 2 and 3 configuration. Figure 4 illustrates the variation in performance between the AVOCA and TCP approach.

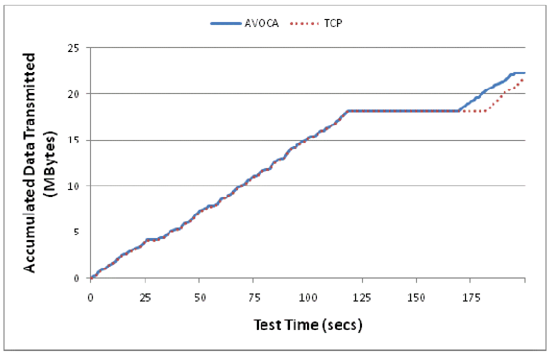

Figure 4. Accumulated Data Transmitted AVOCA/TCP Between 2 Stations

In order to analyse the disparity in throughput we consider the TCP packet transmission times following the network outage between Paddington and Edgware Road. At 120.28 sec the transmission of packet 12859 fails. Following a 2 second time out loss of the packet is detected at 122.28 sec. As a result of the timeout, all in flight packets are discarded, the cwnd is halved and binary exponential back off is implemented doubling the RTO. At 126.28 sec, 134.287 sec and 150.28 sec repeated retransmissions of packet 12859 fail as the train remains out of coverage. At 170 seconds the train enters the coverage of Edgware Road. At 182.28, following a 32 second timeout, the retransmission of packet 12859 is finally successful. Table 4 illustrates the TCP congestion control parameters for the period immediately before and immediately following the network outage.

Time	cwnd	ssthresh
119.2871	3146	3146
119.8348	2936	2936
120.2871	1468	2936
182.2871	1468	2936
182.5193	2936	2936
182.5517	2936	2936
182.5517	4404	2936

Table 4: Congestion Control Parameters During Network Outage

At 120.28sec the size of the sending congestion window is one packet, 1468 bytes. As a connection oriented protocol this packet must be successfully transmitted to the receiving application layer before any subsequent packets can be transmitted. By default TCP packet retransmission only occurs following RTO timeout. Timeouts of packet 12859 occur when the train is out of coverage at 122.28sec, 126.28sec, 134.28sec and 150.28 sec. The train enters the coverage of Edgware Road at 170 sec, however as the cwnd is only 1 packet, transmission cannot be reinitiated until packet timeout occurs at 182.28 sec.

Figure 4 and Table 4 illustrate that TCP performance for vehicle based systems is significantly dependant on the time duration between network coverage areas. This performance deficiency can be explained as the loss in throughput for the period of the remaining RTO timeout. The level of the performance deficiency is bounded by the configurable parameter RTO.Max. The default value of RTO.Max specified in RFC [17] is 60 seconds. For this reason there may be a potential delay of up to 60 seconds before a TCP session reinitiates transmission following Layer 2 and Layer 3 configuration.

IX CONCLUSIONS

In this paper we proposed A Vehicle Oriented Congestion control Algorithm (AVOCA) which considers network coverage holes when implementing congestion control. We illustrate that the congestion control procedures implemented in connection oriented transport layer protocols such as TCP and SCTP are unsuitable for modern vehicle systems in which network connectivity is intermittent in nature. We present a new form of congestion window blocking communication failure and illustrate that existing mechanisms fail to fully utilise intermittent connectivity as the vehicle enters zones of coverage. We illustrate that in a worst case scenario communication will be delayed by up to the value specified by RTO.max; by default 60 seconds. Results presented illustrates that the AVOCA approach removes the new form of congestion window blocking and improves throughput by up to 42% while maintain Internet fair usage policies.

REFERENCES

[1] Institute of Electrical and Electronics Engineers, “IEEE Standard for Local and metropolitan area networks – Part 21: Media Independent Handover Services”, 21 Jan. 2009.

[2] NIST, “The Network Simulator NS-2 NIST Add-on – Neighbor Discovery”, January 2007

[3] R. Stewart, “Stream Control Transmission Protocol (SCTP) Packet Drop Reporting“, draft-stewart-sctp-pktdrprep-09.txt, December 2009

[4] T. Dyer , Rajendra V. Boppana, **“**A comparison of TCP performance over three routing protocols for mobile ad hoc networks”**,** Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking and computing 2001

[5] Y. Takemoto, J. Funasaka, S. Teshima, T. Ohta “SCTP performance improvement for reliable End-to-end communication in ad hoc networks” International Symposium on Autonomous Decentralized Systems, 2009.

[6] Fallon, S, Jacob, P, Qiao, Y, Murphy, L, “An Adaptive Optimized RTO Algorithm for Multi-homed Wireless Environments” in proceedings of the 2009 Wired and Wireless Internet Communications (WWIC) conference, Lecture Notes in Computer Science (LNCS) Volume 5546/2009, pages 133-145

[7] M. Sepulcre, J. Gozalvez, “On the Importance of Application Requirements in Cooperative Vehicular Communications” Wireless On-Demand Network Systems and Services (WONS), 2011

[8] He, S. Chen, S. Chen, T. Cheng, W. “Adaptive congestion control for DSRC vehicle networks” IEEE Communications Letters, Volume 14 Issue 2, February 2010

[9] J. Harri and H. Hartenstein, “Contextual Communications Congestion Control for Cooperative Vehicular Networks”, in IEEE Transactions on Wireless
Communications, 2011.

[10] V. Paxson, M. Allman, Computing TCP's Retransmission Timer, RFC 2988, 2000

[11] Karn, P. and C. Partridge, "Improving Round-Trip Time Estimates in Reliable Transport Protocols", SIGCOMM 87.

[12] Jacobson V.Braden R. Borman. D “TCP Extensions for High Performance” May 1992

[13] Jacobson, V., "Congestion Avoidance and Control", Computer Communication Review, vol. 18, no. 4, pp. 314-329, Aug.

[14] Allman, M. Paxson, V. “On estimating end-to-end network path properties**”** Proceedings of the conference on Applications, technologies, architectures, and protocols for computer communication 1999 Pages: 263 - 274

[15] UC Berkeley, LBL, USC/ISI, and Xerox Parc: ns-2 documentation and software, Version 2.29, Oct. 2005, www.isi.edu/nsnam/ns.

[16] Public Transport Statistics Bulletin GB: 2009 edition available from ttp://www.dft.gov.uk/pgr/statistics/datatablespublications/public/

[17] Transmission Control Protocol RFC 793 http://tools.ietf.org/html/rfc793