# Chapter 12

# Conclusion

## 12.1 Introduction

This thesis has described the design and testing of a rate-based framework for controlling network congestion in connectionless packet-switched networks.

Network congestion occurs when the demands on the resources of a network are too great; typically this results in increasing queue sizes in routers, causing packet delays and eventually packet loss. In the networks under consideration, congestion control attempts to partition network resources and control the sources of network traffic so that the burden placed on the network is sustainable and does not cause congestion.

Traditional congestion control mechanisms that are end-to-end based (such as TCP) have typically relied on indirect evidence to determine if, and how, a network is congested: loss of packets and changes in round-trip times. These mechanisms have been shown to be poor both at controlling congestion, and at partitioning network resources fairly.

Other techniques such as packet queueing and packet loss schemes have been suggested to alleviate some of the problems with the end-to-end schemes. However, these can only be employed when portions of the network are already congested.

Newer congestion control mechanisms such as DecBit and Source Quench have been proposed which feedback explicit information about network congestion to the source. As they are binary-feedback schemes, they take several round-trip iterations to bring each source to a transmission level which gives peak network power.

Alternatives to these traditional forms of congestion control are techniques which are rate-based. Given that oversubscribed network resources are typically links with a fixed bit rate, it should make sense to deliver a value for a sustainable transmission rate to each source. Transmitting at this sustainable rate, a source should not congest the network. Such explicit rate feedback schemes as EPRCA have been adopted in connection-style networks such as ATM. The application of such techniques in connectionless packet-switched networks might overcome some of the drawbacks of traditional control schemes. However, this would require adaptation of the techniques to overcome the substantial differences in network architecture.

## 12.2 A New Rate-Based Control Framework

Given the problems of traditional congestion control mechanisms and the promise of rate-based techniques, I proposed a rate-based congestion control framework for connectionless packet-switched networks which should remedy these problems. The design considerations for the framework, given in Chapter 4 were that:

- Transport congestion control should use congestion information from routers,
- Dropping packets is highly undesirable,
- Router queue sizes >1 are highly undesirable,
- Use of round-trip timers is undesirable,
- Use of sliding windows for flow and congestion control is undesirable,
- Packet retransmission should be done only if necessary,
- Separation of error control and flow/congestion control is desirable, and
- Packet admission and readmission should be smooth.

The resulting framework consists of a rate-based Transport Layer whose rate is controlled by congestion information obtained from all routers along a traffic flow's path. The routers employ a distributed resource allocation algorithm to determine this rate, known as a *sustainable rate*. To do this, each router must monitor the set of traffic flows which it is routing; specifically, the set of traffic flows on each output interface.

The framework, thus, is a combination of operations performed by the end-host (i.e the Transport and Network Layers) and the intermediate routers (i.e the Network Layer). At the end-host, the two layers intercommunicate congestion information as described in Section 7.4. Within the network proper, all packets carry a number of congestion fields:

**Desired_Rate:** The bit rate which the source of this traffic desires.

**Rate:** The bit rate which has been allocated to this flow of traffic by a router.

**Bottle_Node:** The router which set the value of the **Rate** field in the packet.

**Return_Rate:** The value of the last **Rate** field which reached the destination.

Other fields indicate to routers when traffic flows terminate.

The framework operates as follows:

1. The Transport Layer uses a rate-based congestion control scheme, using rate information provided by the Network Layer.
2. A source admits packets for each traffic flow into the network uniformly spaced in time.
3. The routers in the Network Layer implement a congestion control scheme, part of which measures the sustainable traffic rate across the network for each traffic flow: this is performed by the Sustainable Rate Measurement function. This sustainable rate measurement is passed to the destination machine and then via acknowledgment packets to the source machine.
4. Routers implement both congestion avoidance and congestion recovery mechanisms as the rest of the Network Layer congestion control scheme. Routers partition the available resources fairly amongst traffic flows, in both uncongested and congested operating regimes.

## 12.3 Framework Functions

The proposed framework consists of a number of functional elements. Some functions, such as packet dropping mechanisms, can be implemented with existing algorithms. Two elements, that of the Transport Layer function, and that of the Sustainable Rate Measurement function, had to be instantiated.

A transport protocol, TRUMP, which satisfied the requirements for a Transport Layer, was described in Chapter 5. Although this protocol was specifically designed to complete the proposed congestion control framework, it has been specified and verified in enough detail to be considered a 'real' and implementable protocol. TRUMP provides a reliable unidirectional connection from a source application to a destination application across an unreliable network. Features of the protocol include:

- Error detection and recovery using selective acknowledgment and retransmission.
- Explicit and implicit connection setup and connection teardown.
- Rate-based flow and congestion control, using information passed to TRUMP by the Network Layer.
- An infinite 'buffer' size: the source can transmit all of a message before any of the acknowledgments have arrived. This aids high throughput.
- Only one timer required per message during data exchange, which is in the source. The timer is not based on the round-trip time.
- Up to 16 possible types of transport header/data encryption. The encryption covers all fields except for the protocol version number, the encryption type and the segment checksum.

Although TRUMP is a complex protocol, its correctness was partially proven using the Spin protocol verification tool in Appendix B.

An algorithm which meets the requirements of a Sustainable Rate Measurement function, RBCC, was presented in Chapter 6. RBCC is an algorithm distributed across all routers of the network, which calculates a set of sustainable rates for all flow sources to achieve peak network power. The design goals for RBCC were that:

- The function should allocate enough bandwidth to meet each traffic flow's desired rate, if possible.
- The function should allocate bandwidth to flows so that no output interface's available bandwidth is exceeded.
- If a router is a bottleneck, the function should communicate its allocated rates to each source affected, and to other routers.
- The function should allocate bandwidth to flows, taking into account any bottlenecks upstream (towards the source) and downstream (towards the destination).
- Flows should be free to change their desired rate at any time and to any value. The function should cope with changes in a traffic flow's desired rate.

RBCC uses a number of heuristics, to determine if a packet received at a router should cause a recalculation of the set of sustainable rates allocated by the router, held in an Allocated Bandwidth Table (ABT). This recalculation is performed by an algorithm which must meet several criteria:

- Allocation of rates amongst the flows in the ABT must be fair.
- A traffic flow's rate allocation must not exceed the desired rate of its source, nor may it exceed the rate set by a bottleneck router which is not this router.
- A traffic flow's rate allocation cannot be set to zero or to $\infty$.
- The sum of allocated rates in the ABT must not exceed the output interface's bit rate.

The proposed algorithm is given in Section 6.2.3. Various issues with the framework were raised in Chapter 7: the effectiveness of the framework as a feedback loop; the identification of traffic flows within the network; the required congestion fields within packets; the intercommunication of congestion information between network layers, and the complexity and overhead of the framework itself. Some issues raised still need further examination.

## 12.4 Experimental Results

The proposed rate-based congestion control framework was tested experimentally using a network simulator, for two main reasons:

1. Theoretical analysis of the framework as a control scheme would limit its validation to very simplistic network architectures, and
2. It was desirable to observe the frameworks control of congestion on a number of realistic network scenarios. Experimental testing would also reveal the sensitivity of the framework's parameters on its congestion control performance.

Experiments described in Chapter 9 concentrated on network scenarios which were hand-crafted to reveal any deficiencies with the control of congestion, and the fairness, in the framework. The framework was also compared with two traditional end-to-end control schemes, TCP Reno and TCP Vegas.

The hand-crafted network experiments showed that the proposed framework appeared to deal with network congestion more than adequately: no packets were lost in any of the experiments, and queue lengths in routers stayed close to one packet queued. Overall, link utilisation was high, and end-to-end variance was low. The framework controlled congestion, and allocated rates to all sources in a fair manner. In comparison, TCP Reno and TCP Vegas achieved poorer congestion control, with large packet loss, high router queue lengths, lower link utilisation and higher end-to-end variance.

In order to give a more statistically valid experimental analysis of the framework, 500 network scenarios were pseudo-randomly generated. Each scenario had 15 traffic flows, 9 routers, and an arbitrary network topology. Link capacities were arranged so as to increase the likelihood of congestion conditions.

The proposed framework was simulated within each of the 500 network scenarios, and the results described in Chapter 10. Again, TCP Reno and TCP Vegas were also simulated to provide a comparison with traditional control techniques. The results confirmed those obtained with the hand-crafted scenarios: the proposed framework controlled congestion very well, and allocated rates to all sources in a fair manner.

The effect of specific parameters within the TRUMP and RBCC components of the framework was examined in Chapter 11. Most of the parameters had little effect on the framework's congestion control performance, except for extremes in parameter values. Two of the five parameters examined, the threshold $T$ and the scaling factor $\alpha$, had a significant negative impact on the framework's congestion control operation for certain values. A set of parameter values was recommended, and their effect on the framework was shown to have a positive effect on congestion control performance.

## 12.5 Future Work

During the course of this work, several issues with the proposed congestion control framework have been raised but left unanswered: many issues were discussed in Chapter 7. The constraints on the size, duration and scope of this research have prevented me from exploring these issues. They should provide several areas for future research into rate-based network congestion control.

The suitability of the proposed framework as a congestion control mechanism has only been experimentally verified. Rigorous, mathematical analysis of the framework as a distributed algorithm should help to confirm its worthiness.

Two main functions in the framework are the Transport protocol, TRUMP, and the Sustainable Rate Measurement function, RBCC. TRUMP as a protocol was not specified enough to be fully implemented in a real protocol stack. Its specification should be completed, and an implementation of TRUMP created for a suitable protocol stack. Similarly, RBCC should be implemented as part of a real protocol stack.

Although the thesis has shown RBCC implemented within routers, in a real network it must also be implemented within sources of traffic flows: this arises as typical source machines must multiplex many independent application data flows onto one (or more) network interfaces. RBCC within the source would ensure that the network interfaces' bandwidth is shared fairly amongst the application data flows. An implementation of RBCC would have to be done for sources as well as routers.

A combined TRUMP/RBCC implementation would allow studies of the proposed congestion control framework in real-life network scenarios. This would verify the results of the network simulations given in this thesis, and allow the effects of Link Layers and hardware on the framework to be examined. A real-life implementation of RBCC would facilitate the study of the computational overhead of the the framework on routers, and provide the incentive to improve its implementation efficiency.

Section 5.3.2 describes TRUMP's use of a lazy retransmission strategy to avoid the setting of any round-trip timer. This can cause excess network load when network capacity is high and round-trip delays are large. An alternate method of prompting the destination for acknowledgments should be studied, and its effect upon network load should be compared to lazy retransmission.

The constitution of a flow of traffic within the framework was discussed in Section 7.2, and resolved to be the flow of data between two applications. This does add a burden on the framework in comparison to other alternatives discussed. The feasibility and overhead of the other flow identify alternatives should be examined.

The framework deals with route changes in a primitive fashion. Further studies should be conducted to allow the framework to react more intelligently. A synergy may be achieved if the congestion control framework was coupled with the network's route control framework: this needs to be investigated.

Within RBCC itself, several heuristics were described: the $T$, $\alpha$ congestion avoidance mechanism and the ABT update decisions. A fruitful area of research would be to examine these heuristics and replace them with ones that improve the congestion control of the framework. The congestion avoidance mechanism is particularly primitive, and deserves special attention.

On a higher level, both TRUMP and RBCC are only particular function instantiations which can be used within the congestion control framework. Other Sustainable Rate Measurement function instantiations may provide better congestion control within the framework than RBCC. Certainly, Transport Layers other than TRUMP are required to provide particular value-added services to particular applications.

The Fair Share rate allocation scheme [Charney 94] should be investigated to determine if it can be used as is, or modified, to work as a Sustainable Rate Measurement function within the proposed framework. If so, its rate allocation performance and overhead should be compared with RBCC.

Communication within the framework components is performed by a number of congestion fields in each packet, and infrequently by the exchange of Rate Quench packets. The number and types of fields can be changed to provide such things as flow priorities and congestion-oriented qualities of service. This should be studied. The trigger for Rate Quench packets, which network components may send them, and their effect upon sources, also deserves further study.

## 12.6 Summary

The early adoption of sliding windows for flow control in connectionless packet-switched networks precluded the use of rate-based techniques to control congestion. Sliding window flow control was modified to perform end-to-end congestion control, but several problems (round-trip estimation, bursty packet admission, window side-effects) render congestion control with these techniques difficult.

An alternative approach is to use rate-based techniques for congestion control. Such techniques have been adopted for congestion control in other network architectures. This thesis has examined the requirements for a rate-based congestion control framework for connectionless packet-switched networks. From these requirements, a framework has been designed, and the functionality (and design criteria) for its components has been instantiated. Two major components — a rate-based Transport protocol, TRUMP, and a Sustainable Rate Measurement function, RBCC — were described.

In order to verify the effectiveness of the congestion control framework, it was analysed experimentally in a large number of simulated network scenarios. The parameters of the framework's components were analysed, and the framework congestion control performance was compared with traditional Internet control mechanisms.

The rate-based control framework was shown to excel at congestion control, giving a packet loss result orders of magnitude lower than the traditional control mechanisms. Results for router queue lengths, link utilisation, rate allocation and end-to-end variance confirm the congestion control performance of the framework.

The framework has been designed to be modular, and to allow the individual framework components to be modified or replaced, as long as the framework's operational requirements are met. Further research may see improved components, or new components which enhance the effectiveness of the framework's congestion control, or reduce the overhead of the framework on the network's components.