Combinatorial Design of Multi-Ring Networks with Combined Routing and Flow Control [1]

Yueyue Song [2] and Avishai Wool [3] and Bülent Yener [4]

Abstract

In this paper we present a novel design technique for packet switched networks. The design is based on the construction of multiple virtual rings, which enjoy the *one-bridge property*: the path between any two nodes is either confined to a single ring, or traverses exactly two rings (passing through a single bridge node). Our best designs are constructed by using Finite Generalized Quadrangles of combinatorial design theory. We present novel routing and flow control protocols that capitalize on the one-bridge property of the multi-ring network. Our protocols ensure that (i) no loss due to congestion occurs inside a network, under arbitrary traffic patterns; (ii) all the packets reach their destinations within bounded time with low jitter; and (iii) the bandwidth is allocated fairly and no host is starved. We provide both a theoretical analysis and an extensive simulation-based performance evaluation of our protocols.

Key words: Ring networks, Generalized Quadrangles, Congestion-free

---

[1] Parts of this paper were presented at the Globecom'99 conference, Rio De Janeiro, 1999.

[2] `yueyue@cs.rutgers.edu`. Supported in part by the Center for Discrete Mathematics and Theoretical Computer Science (DIMACS) which is funded by the National Science Foundation under contract STC-91-19999.

[3] Dept. of Electrical Engineering Systems, Tel Aviv University, Ramat Aviv 69778, Israel. `yash@acm.org`.

[4] Computer Science Department, Rensselaer Polytechnic Institute, 110 8th Street, Troy, NY 12180-3590, USA. `yener@cs.wpi.edu`.

Preprint submitted to Computer Networks *1 October 2002*

1 Introduction

1.1 Overview

The ever-increasing growth of data traffic and its bandwidth demand necessitates careful design and planning of the infrastructures of next generation networks. Traditional approaches to network design either face computationally hard optimization problems, or use heuristic methods with approximate answers. The complexity increase in such approaches is partially due to dependency to the traffic models and estimations. In contrast, combinatorial approaches to network designs [Var86,YOY94,YOY97] offer deterministic bounds on the maximum route length, and on the survivability, independent of the traffic characteristics. Ensuring traffic-independent properties is particularly important for data networks where traffic is bursty and unpredictable.

As the starting point of this paper, we introduce a new class of virtual-ring network designs. These networks enjoy the *one-bridge property*: the path between any two nodes is either confined to a single ring, or traverses exactly two rings (passing through a single bridge node). Our best designs are constructed by using *Finite Generalized Quadrangles* of combinatorial design theory.

Our goal is to show which routing, flow and access control protocols are best suited to such network topologies. We consider two basic approaches. With a unified approach, the protocols are tightly coupled with the network design, so they can take full advantage of its topological properties. With a decoupled approach, the protocols are standard WAN protocols, which are oblivious to the network design. Instead, they employ shortest path routing and window-based flow/congestion control. This paper tests the two alternatives and compares their performance.

1.2 Background and Related Work

We focus on the ring topology as a basic building block for several reasons. The ring topology is well studied, and is widely used in LAN and MAN environments. Its attractive symmetry and cyclic structure allow for simple, decentralized, fair, and congestion-free control protocols. And since a ring is a minimal biconnected graph it also provides survivability with minimal cost. For example, a multi-ring topology is the backbone topology of choice in the Synchronous Optical Network (SONET) standard [ANS88]. Surveys on other uses of multi-ring networks are [DH98,LSBW98].

Recently there has been a renewed interest in ring-based networks in the context of Resilient Packet Rings (RPR) networks [RPR01,CHG$^+$01]. RPR is a new data transport technology for Metropolitan Area Networks (MAN), to be standardized as IEEE 802.17. An RPR is a ring-based architecture that consists of two counter-rotating rings. RPR generalizes the spatial bandwidth reuse by adapting mechanisms found in buffer insertion rings.

Networks with a simple linear topology, such as a bus or a ring [FN69], [MB76], [BCK$^+$83], are not throughput scalable. In contrast, networks with an arbitrary topology may be throughput scalable but they are typically not congestion-free. The only exception we are aware of is the MetaNet architecture [OY90,OY95] which provides ring properties on a LAN with an arbitrary topology, by embedding a virtual ring around a spanning tree of the network. The virtual ring emulates a full-duplex ring with spatial bandwidth reuse, like the MetaRing [CO93,Ofe94]. Since the ring spans every node, the maximum length of routing is $O(N)$ for an N-node network. Thus, scaling the MetaNet with a single virtual ring to a wide area network (WAN) may not be efficient.

Combinatorial design theory was first applied to multi-ring network designs in [YOY94,YOY97]. The authors showed how to use *Balanced Incomplete Block Designs* (BIBDs) to obtain congestion-free networks with scalable throughput. The techniques resulted in networks in which the maximum route length and the maximum degree (number of rings a node belongs to) are both bounded by $O(\sqrt{N})$, where N is the number of nodes in the network. These bounds are similar to those of earlier approaches to multi-ring network design (e.g., chordal rings [RGA85,RS86] or ring-connected-ring [DP91]), with the additional property of congestion-free routing.

1.3 Contributions

Our first contribution is the introduction of network designs with the one-bridge property. We prove a new trade-off between the node degree and the ring size for such networks. Then we introduce Generalized Quadrangles (GQ) and GQ-based network designs. Our GQ-based networks have a high level of path redundancy which translates to high survivability in the face of link and node failures, and allows easy load balancing among rings. And our networks require only $O(N^{4/3})$ links in total, which significantly improves the $O(N^{3/2})$ links of [YOY94].

Our next contribution is a suite of new topology-aware protocols which we collectively call the *VRing* protocols. These protocols are tailored to virtual multi-ring networks which have the one-bridge property. Within each ring our protocols emulate a virtual slotted ring [LOY96,BOY97]. Our routing and control protocols ensure that (i) the network is congestion-free, under arbitrary traffic patterns; (ii) all the packets reach their destinations within bounded time; (iii) the bandwidth is allocated fairly and no host is starved.

We evaluated the performance of our protocols by an extensive delay and throughput simulation study. We first tested the behavior of our topology-tailored *VRing* protocols on their own. Then, we compared the performance of *VRing* to an Internet-like suite of protocols which we call the *INet* protocols.

Our results indicate that the network access delay is substantially higher in the *VRing* than in the *INet*. However, once the packets enter the network, in the *VRing* they are routed with no loss, while in the *INet* they may be dropped and retransmitted multiple times due to congestion. Thus, our results show that the end-to-end network delay in the *VRing* is significantly lower than in the *INet*. Furthermore, we show that the variability of the network delay (and hence the jitter) in the *VRing* is quite small, and remains bounded even when the traffic intensity increases. In contrast, the network delay has large variability in the *INet*, and the variability grows rapidly with the traffic intensity. Finally, our results show that the network links are utilized more efficiently in the *VRing*.

We conclude that from the network provider's perspective, the *VRing* provides much better network utilization, and allows the provider to give the network's users QoS guarantees regarding available bandwidth, fairness, and jitter. Furthermore, for user applications that are more sensitive to jitter than to delay, such as audio and video transmissions, the *VRing* approach may be advantageous.

Organization: The rest of this paper is organized as follows. In Section 2 we describe the network design model, and introduce generalized quadrangle designs. Sections 3 and 4 describe the *VRing* access control and flow control protocols. In Section 5 we analyze the *VRing* protocols' properties. In Section 6 we present the results of a simulation-based performance evaluation of both the *VRing* and the *INet*. We conclude in Section 7.

2 Combinatorial Construction of Rings

The basic design criterion in all our multiple ring network designs is that they should obey the one-bridge property. By this we mean that a packet would need to cross at most one ring-to-ring bridge along its path from any node to any other node in the network. We shall see that this property lets us design networks with small rings (giving low propagation delay) and low degree—and at the same time admits very efficient congestion-free flow-control protocols. Formally we use the following definition.

Definition 2.1 *Let $\mathcal{R} = R_1, \ldots, R_M$ denote the rings of a multi-ring network. Then $\mathcal{R}$ is said to have the one-bridge property if for every two nodes x, y one of the following conditions holds:*

(1) There exists a ring R_i such that $x \in R_i \ni y$. We say that x and y are neighbors *on ring R_i.*
(2) There exist two rings R_i, R_j and a node z such that $x \in R_i \ni z \in R_j \ni y$. In this case we call z the bridge node*, or simply the bridge.*

Clearly block-designs such as those used in [YOY94] have the one-bridge property in a trivial way: every two nodes are neighbors on some ring, so condition 1 of Definition 2.1 always holds. In this paper we focus on designs in which many (indeed, the vast majority) of the node pairs are *not* neighbors on any ring.

2.1 Generalized Quadrangles

Generalized Quadrangles (GQs) are a rich class of combinatorial designs that enjoy the one-bridge property of Definition 2.1. A succinct survey of these combinatorial objects can be found in [CD96, ch. IV.21]. An in-depth mathematical treatment of GQs can be found in the monograph [PT84]. Even a brief introduction to the theory of GQs would exceed the scope of this paper, so we limit ourselves here to the very basic properties of GQs, with emphasis on how they fit into our methodology of network design.

Definition 2.2 [CD96, IV.21.1] *A* finite generalized quadrangle (GQ) *is a collection $\mathcal{R}$ of sets (rings) over a universe U of nodes satisfying the following:*

1. *each node belongs to $t+1$ rings, and two distinct nodes belong to at most one ring together;*
2. *each ring contains $s+1$ nodes, and two distinct rings have at most one node in common;*
3. *if $x \in U$ is a node and $R \in \mathcal{R}$ is a ring such that $x \notin R$ then there exist a unique $y \in U$ and $S \in \mathcal{R}$ such that $x \in S \ni y \in R$.*

A GQ *with parameters s, t is said to be of* order *(s, t).*

The next proposition shows that not only do the GQs have the one-bridge property, they have additional symmetry properties that are useful for our routing protocols.

Proposition 2.3 *Let x, y be nodes in a* GQ *of order (s, t). If x and y are not neighbors then there exist $t+1$ distinct pairs of rings R_i^x, R_i^y such that $x \in R_i^x$, $y \in R_i^y$, and $|R_i^x \cap R_i^y| = 1$ for all $i = 1, \ldots, t+1$.*

{1,4,5}	{2,4,6}	{3,4,7}	{5,10,15}	{6,11,13}
{1,8,9}	{2,8,10}	{3,8,11}	{5,11,14}	{7,9,14}
{1,12,13}	{2,12,14}	{3,12,15}	{6,9,15}	{7,10,13}

Table 1
The $W(2)$ generalized quadrangle design, with $N = 15$ nodes of degree $d = 3$, and $M = 15$ rings of size $n = 3$.

PROOF. By definition, x belongs to $t+1$ rings, which we denote by R_i^x for $i = 1, \ldots, t+1$. The node y does not belong to any of these R_i^x rings since x, y are not neighbors. So by condition 3 of Definition 2.2, for every such R_i^x there exists a unique R_i^y which contains a single bridge node z_i such that $x \in R_i^x \ni z_i \in R_i^x \ni y$. $\square$

Remark: Proposition 2.3 proves much more than the one-bridge property. It in fact shows that if x needs to send a packet to y and they are not neighbors, then x can select *any* of the $t+1$ rings it is a member of and place the packet on this ring. The proposition shows that on each of x's rings there exists a bridge that also belongs to one of y's rings and is capable of bridging the packet to its destination.

Proposition 2.

Name	Nodes N	#rings	degree d	ring size n	neighbors
$W(2)$	15	15	3	3	6
$Q(5,2)$	27	45	5	3	10
$W(3)$	40	40	4	4	12
$Q(5,2)$-dual	45	27	3	5	12
$\mathcal{S}(\Omega^-,4)$-dual	64	96	6	4	18
$W(4)$	85	85	5	5	20
$\mathcal{S}(\Omega^-,4)$	96	64	4	6	20
$Q(5,3)$	112	280	10	4	30
$\mathcal{S}(\Omega^-,5)$-dual	125	175	7	5	28
$W(5)$	156	156	6	6	30
$H(4,2^2)$	165	297	9	5	36
$\mathcal{S}(\Omega^-,5)$	175	125	5	7	30
$Q(5,3)$-dual	280	112	4	10	36
$H(4,2^2)$-dual	297	165	5	9	40
$Q(5,4)$	325	1105	17	5	68
$\mathcal{S}(\Omega^-,7)$-dual	343	441	9	7	54
$W(7)$	400	400	8	8	56

Table 2
The parameters of known GQ constructions, with up to $N = 400$ nodes.

In order to construct a GQ-based network of N nodes with degree d we need a GQ of order (s, t) that has N nodes such that $d = t + 1$. However, Proposition 2.4 shows that a GQ always has the constraint $N = (s + 1)(st + 1)$. Therefore not many integers N are suitable as GQ sizes. Moreover, GQ designs are not even known for all N's that *do* satisfy the constraints. Thus, the number of GQ designs that we can apply directly is quite limited. To construct networks of arbitrary size, we use the methods of [YOY97]. Briefly, the approach is to find a GQ(s, t), which has fewer than N nodes and in which the degree bound d is not exceeded (i.e., $t+1 \le d$). This GQ is scaled up to N nodes by using two scaling operations: insertion and multiplication. We refer the reader to [YOY97] for more details.

2.2 A Trade-off between Ring Size and Node Degree

The one-bridge property constrains the characteristics of the achievable designs. In particular there is a trade-off between the ring size n and the node degree d on an N-node network. However, the trade-off imposed by our one-bridge property is much better than that of [YOY94]: their designs had $nd = \Omega(N)$ so n and d could achieve at most $O(\sqrt{N})$ simultaneously. In contrast, we have $nd = \Omega(\sqrt{N})$ so we can hope to achieve n and d of $O(N^{1/4})$ simultaneously.

Theorem 2.5 *Consider a multi-ring network over N nodes that has the one-bridge property. Let n denote the size of the largest ring and let d denote the maximal node degree. Then*

$$nd = \Omega(\sqrt{N}).$$

PROOF. Consider some node u. This u has at most $d(n-1)$ distinct nodes as neighbors. Each of these neighbors belongs to at most $d-1$ rings which u does not belong to, so u has at most $d(n-1)(d-1)(n-1)$ distinct neighbors-of-neighbors. By the one-bridge property, every node $v \neq u$ is either a neighbor or a neighbor-of-a-neighbor of u. Hence

$$N \leq 1 + d(n-1) + d(n-1)(d-1)(n-1) = (nd)^2(1 + o(1)). \qquad \square$$

Remark: Theorem 2.5 is not tight for GQs, which have $N = O(n^2 d)$. In fact, we are not aware of any family of combinatorial constructions with the one-bridge property for which Theorem 2.5 is tight. Closing the gap between the lower and upper bounds is an open question. We conjecture that it may be possible to find non-GQ constructions with the one-bridge property, for which $N = O((nd)^2)$: This is because, as we remarked after Proposition 2.3, GQs are much more constrained than the one-bridge property requires.

2.3 Redundancy, Self-Routing, and Cost

A GQ-based network of degree d has a high degree of path redundancy: Proposition 2.3 shows that there are d edge-disjoint one-bridge paths between every two non-neighboring nodes u and v. In fact u can choose to send a packet $P(v)$ to v on any ring $R_i \ni u$ it belongs to, and a bridge $x \in R_i$ is guaranteed to exist to relay the packet onto a ring R_j that v belongs to.

This combinatorial property gives us a "self-routing" capability: In principle, the source u does not really need to know how to route a packet to a destination v that is not a neighbor, or even to know the identity of the bridge node x—the packet can be placed on any arbitrary ring, and it will arrive at v.

Using a simplistic model of cost we can see that GQ-based networks achieve their desirable properties quite cheaply. If we associate a unit of cost for each link, then a GQ-based network with $M = N$ rings of size $O(N^{1/3})$ would have a cost of $O(N^{4/3})$. This is only slightly more than the minimal cost of $N-1$ that is required for basic network connectivity, and is less than the $O(N^{3/2})$ cost of the BIBD-based networks of [YOY94].

3 The *VRing* Protocols: Definitions and Access Control

We now start to present our topology-aware *VRing* protocols. The *VRing* protocols have two components: access control, and flow control. In this section we describe the model we use, introduce our access control protocol, and analyze its properties. The flow control protocol is described in the next section. As we shall see, the combination of the access and flow control protocols guarantees congestion-free and fair routing with bounded network delay, while using bounded internal buffers.

3.1 Network Model and Assumptions

(1) Each ring has a unique ID R_i. Each node u has a virtual ID (*VID*) which is the set of its virtual ring IDs (e.g., if a node belongs to rings R_1, R_3, R_7, then $VID(u) = \{R_1, R_3, R_7\}$). [5]
(2) Each node u knows the *physical* address v and the *VID* of each of its ring-neighbors on every ring it belongs to.
(3) Each packet header carries the physical address of its destination. Using the information in 1 and 2 , each node u can perform a routing-table-check function for a given destination v to determine the virtual ID of destination v.
(4) A packet $P(v)$ sent from source u to destination v is called *LOCAL* if $VID(u) \cap VID(v) \neq \emptyset$ (i.e., u and v are neighbors on some ring). Otherwise $P(v)$ is called *REMOTE*.
(5) For each *REMOTE* packet $P(v)$ originating at u, u is able to select a ring R_i for $P(v)$ which contains the *bridge* node x guaranteed by the one-bridge property. Namely, $R_i \in VID(u) \cap VID(x)$ and also $VID(x) \cap VID(v) \neq \emptyset$.
(6) The network is synchronized (e.g., using the global positioning system (GPS) [Kap96,Dan98]) and time is discretized to time slots. We assume that the slots rotate clockwise at each clock tick and the time interval between two consecutive ticks is long enough to both remove and insert a packet.
(7) Each slot has a 1-bit status field marking the slot as being either *FULL* (i.e., carrying a packet) or *EMPTY*. We denote the time it takes for a slot to rotate all the way around the ring by D, the link data rate by C, and the number of slots (packets) simultaneously on a ring by p. So for

---

[5] For simplicity we assume that the network is completely symmetric. In particular we assume that all the rings have the same number of nodes, denoted by n, and all the nodes have the same *degree* d (i.e., each node belongs to exactly d rings). Note that the GQ-based constructions we described in Section 2 indeed enjoy this symmetry.

packets of size s the link data rate must obey the inequality $C \geq \frac{ps}{D}$. [6]

(8) Nodes forward packets according to the *pseudo-isochronous* or *RISC-like* forwarding principle proposed in [LOSS94,LOY96]. Thus, a packet received by a node in slot t, is forwarded by the node so that it reaches the next node along the path in slot $t + i$ (for some $i \geq 1$).

3.2 Quota-based Access Control

The purpose of our access control is to ensure fairness among different nodes. For this we introduce a quota which we denote by k, and allow a node u to transmit up to k packets on each ring R_i that it belongs to, during any cycle of duration D. The naive choice is to set $k = \lfloor p/n \rfloor$, and to statically reserve k slots for each node on each ring it belongs to; a node u would be allowed to insert packets only to these designated slots. However, this approach is rather inflexible, and may introduce unnecessary delays, e.g., when a node has a burst of k packets to send, and it sees *EMPTY* slots, yet these slots belong to some other node $v \neq u$. Furthermore, we shall see that we can in fact use

$$k = \lfloor p/(n-1) \rfloor. \tag{1}$$

We shall show that this setting of k does not sacrifice fairness. We can easily increase k further, if we want to use statistical multiplexing (but this may lead to some unfairness if some nodes behave greedily).

To achieve this, we suggest a more flexible implementation which is based on quota counters. A node u holds a quota-counter for each ring it belongs to, and the counter is replenished to k after each cycle. Node u can send packets as long as the counter is positive, and the counter is decremented for every packet sent.

However, we need to be careful about how the quotas are replenished. A naive implementation of a quota-based access control could lead to unfairness, as the following example shows.

Example 3.1 (Unfair quota implementation) Suppose that the nodes on R_i replenish their quotas to k at times $t = 0, D, 2D, \ldots$ (i.e., at the beginning of each cycle). Then it may be possible for node u to send k packets on the last k slots it sees in the cycle which ends at time t, and k more on the first k slots it sees in the next cycle, without violating its quota in either cycle.

---

[6] We assume that one of the nodes (say the one with minimum ID number) in each ring is assigned as the *virtual ring bandwidth manager*. The manager node is the origin of the ring and the slots are numbered from 1 to p according to the time that the origin "sees" them.

However, note that between time $t-k$ and $t+k$ node u used $2k$ slots. From the point of view of a node v downstream from u, all these slots may appear during a single cycle, which in turn may cause v not to be able to send any packets in that cycle.

To avoid this phenomenon, we need to strengthen what we require of the nodes. This is formalized by the following invariant which we require the protocol to obey. Note that this invariant is very similar to the quota scheme used in Frame Relay networks (cf. [Kes97]).

Invariant 3.2 *Every node u may send at most k packets within* any *consecutive period of D time (p clock ticks).*

When the access control protocol obeys Invariant 3.2 we can prove the following theorem, which both shows that the protocol is fair and gives a bound on the access delay. The proof is straightforward, so we provide a sketch here, and refer the reader to [WY99] for full details.

Theorem 3.3 *If node u has $k' \leq k = \lfloor p/(n-1) \rfloor$ packets to send at time t, then u will succeed in sending all of them by time $t+D$.*

PROOF. (Sketch). We first observe that a packet cannot stay on a ring R_i more than D time. Consider a slot which node u cannot use because it is occupied by a packet sent earlier by node v. Since a packet cannot occupy a slot more than D time, v must have placed the packet in the slot during the last D time. Since v obeys Invariant 3.2, there are at most k such slots on the ring at any time. Since clearly u will never see a slot marked by itself, we conclude that u will be able to send its k' packets. $\Box$

Remarks:

- In [WY99] we show how to maintain Invariant 3.2, using a δ-list of timers.
- Theorem 3.3 does not distinguish between different types of traffic. The k packets per cycle that a node is allowed to send contain both new packets (that the node needs to inject into the network) and bridged packets (which were originated by some other node and are already in the network). So Theorem 3.3 in itself is not yet sufficient to provide bandwidth or delay guarantees to the nodes; these would also depend on the flow control protocols used for the different traffic types.
- If we assume that a node cannot remove a packet and immediately insert a packet into the same slot, then we need $k = \lfloor p/n \rfloor$ for Theorem 3.3 to hold.

3.3 Link Utilization with Access Control

Since we are using an access control scheme to guarantee fairness among the nodes, we are also bounding the maximal link utilization that can be achieved. In this subsection we quantify this bound. In particular we show that if the distribution of destinations on the ring is uniform and the rings are large then the utilization cannot exceed ≈50% when access control is used. The proof implies that no protocol can do better on a ring if the protocol ensures fairness and no-loss due to congestion.

Clearly, if the requirements of *no-loss due to congestion* and *fairness* are relaxed (i.e., if we over-allocate the bandwidth based on statistical multiplexing), the utilization can be driven to a much higher value.

Consider a ring R_i of p slots operating for T cycles. We say that a packet that occupies a slot for one clock tick uses one *slot-tick*. Thus during a cycle of p clock ticks at most p^2 slot-ticks may be used, and during T cycles the total number of slot-ticks is p^2T.

The next two definition capture our notions of link utilization and average path length, which we need for Proposition 3.6.

Definition 3.

Remarks:

- The proposition is a quantified version of the well known trade-off between fairness and optimality (of the utilization, in our case) [Tan96]. Note that it holds for *any* fair access control scheme on a slotted uni-directional ring and is not an artifact of our method.
- If the rings are large (so $n/(n-1) \approx 1$), and the distribution of destinations on the ring is uniform (so $E[PLen] = p/2$), then Corollary 3.7 shows that $Util \leq 0.5$. A utilization close to 1 can only be achieved if $E[PLen] \approx p$, i.e., each node sends packets only to its furthest possible destination on the ring.
- The proof ignores the "boundary" effects, namely, that initially the ring may not be empty, and that packets sent during cycle number T may reach their destinations only after the end of this cycle. It is easy to see that when T is large these effects are negligible.

placed on. Once this routing decision is made, the packet is placed in the appropriate buffer. Each node u has two types of buffers (queues) for host-generated traffic that is designated to a ring R_i:

(1) LQ_i for host-to-ring *LOCAL* traffic. u has one such buffer for each ring it belongs to.
(2) $RQ_{ij}(x)$ for host-to-ring *REMOTE* traffic that is designated to ring R_j via the bridge node x. Thus u has $(n-1)(d-1)$ RQ_{ij} buffers for ring R_i, one for each remote ring R_j that can be bridged to via a neighbor x (and a total of $d(n-1)(d-1)$ such buffers for all the rings that u belongs to).[7]

Note that both LQ and RQ are unbounded queues.

In addition, a node u also needs to bridge ring-to-ring traffic. For this u has a buffer BQ_{ij} in which it stores *REMOTE* packets it bridges from R_i and R_j. Thus u has $d-1$ such BQ buffers for every ring R_i it belongs to, and $d(d-1)$ BQ buffers in total. The BQ buffers are bounded, and their size (in packets) is denoted by $MAXBQ$. We shall prove that the flow control is congestion-free regardless of the value of $MAXBQ$.

The flow control is done using a system of quotas that govern the service rates for the different queues. These quotas are in units of packets that a node u can transmit during a cycle rotation time D. Since *LOCAL* traffic does not pose a buffer-overflow problem we do not need to impose a specific quota on the LQ buffer (other than the general access-control quota k). Thus we have the following quotas: A node u is allowed to transmit at most $r_u^{ij}(x)$ packets per cycle out of buffer $RQ_{ij}(x)$.

The r^{ij} quotas are modified adaptively by the flow control algorithm. Intuitively, as a buffer BQ_{ij} at some bridge node x is filled, the bridge adaptively reduces the quotas $r_u^{ij}(x)$ of all the nodes u on ring R_i. The next sections describe the details of the mechanisms involved.

4.2 The Quota Matrix Mechanism

Consider now a node u in its role as a bridge node between R_i and R_j. To avoid dropping packets we need to control the total arrival rate to the bridge buffer BQ in the current cycle as a function of the BQ size at the end of the previous cycle. Informally, the total arrival rate at u's BQ_{ij} buffer is closely related to

---

[7] In the GQ constructions there is always a single node x bridging between rings R_i and R_j. However, in general there may be several possible bridges, which we emphasize by explicitly noting the bridge node x in the queue name.

the sum of remote quotas on the ring R_i, $\sum_{x\in R_i} r_x^{ij}(u)$, thus decreasing the r^{ij} quotas should decrease the arrival rate at BQ_{ij}. However, the $r_x^{ij}(u)$ quotas are local variables at each node x. Thus a mechanism is needed to inform the nodes when quota adjustments are needed on some ring, and we need to account for the delay in propagating the updated quotas to the nodes on the ring.

The mechanism we propose is based on circulating a *quota matrix* Q^i on each ring R_i. Each node u advertises the total number of packets that it is willing to bridge from R_i to R_j in the entry $Q^i[u, R_j]$. If there is a unique bridge node on R_i for every R_j that can be bridged to, then Q^i has $n(d-1)$ entries in total. We need to ensure that the following *flow control* condition holds when the quota matrix *returns* to node u:

$$\sum_{x\in R_i} r_x^{ij}(u) \leq Q^i[u, R_j]. \tag{2}$$

When a node x receives the quota matrix Q^i on ring R_i, x adjusts its *REMOTE* quotas by setting

$$r_x^{ij}(u) \leftarrow \left\lfloor \frac{Q^i[u, R_j]}{n-1} \right\rfloor \tag{3}$$

for all rings R_j that can be bridged to from R_i. Then x needs to calculate the new advertisements for the number of packets *it* is willing to bridge from ring R_i in the next cycle, update the entries it is in charge of in the Q^i matrix (namely the entries $Q^i[x, R_j]$ for all R_j that x belongs to), and forward the Q^i matrix. It is obvious that condition (2) is met when all the nodes use (3) to update their r^{ij} quotas.[8]

For the purpose of flow control, the circulation of the Q matrix gives a natural definition of cycles, as follows.

Definition

- Upon receiving a *FULL* slot containing $P(v)$:

(1) If $v = u$, remove $P(v)$ from R_i, mark the slot *EMPTY*, go to procedure for *EMPTY* slots.
(2) Else if u is a bridge for $P(v)$, remove $P(v)$ from R_i, mark the slot *EMPTY*, store $P(v)$ in the appropriate $BQ_{ij}(u)$ buffer.
(3) Else ROUTE $P(v)$ on R_i.

- Upon receiving an *EMPTY* slot:

(1) Verify that the access control quota k has not been exceeded.
(2) Select a packet $P(v)$ from LQ_i, some BQ_{ji}, or some $RQ_{ij}(x)$ for which the flow control quota $r_u^{ij}(x)$ has not been exceeded.
(3) Mark the slot *FULL*, send $P(v)$ on R_i.

- Upon receiving the Q^i matrix:

(1) Adjust the $r_u^{ij}(x)$ quotas.
(2) Update the $Q^i[u, R_j]$ entries.
(3) Forward Q^i on R_i.

Fig. 1. The combined routing and flow protocol for node u on ring R_i.

by u. However, the quota $r_x^{ij}(u)$ at node x on the same ring R_i is applied to cycles as perceived by x.

4.3 Updating the Quotas

Consider a particular buffer BQ_{ij} at a node u. Let B_m denote the number of packets that u advertises at the *start* of cycle m (i.e., the value that u places in $Q^i[u, R_j]$), and let $|BQ_m|$ be the number of packets in the BQ_{ij} buffer at the *end* of cycle m Recall that the maximal size of $|BQ_{ij}|$ is $MAXBQ$. The following rule is used to update the new advertisement.

$$B_{m+1} \leftarrow MAXBQ - |BQ_m| - B_m, \tag{4}$$

which is simply the difference between the current free space and number of packets committed to at the start of the previous cycle. The combined access control, flow control and routing protocol is sketched in Figure 1.

4.4 The Packet Selection Policy

The quotas of the buffer service rates are not meant to guarantee fairness among the various traffic types that a node handles. Furthermore, when a node u has packets ready in several of its LQ, RQ, and BQ buffers, it still has a degree of freedom in choosing which queue to service in the next clock tick.

Our protocols are mostly indifferent to the design choice made regarding this issue, as long as the policy is *fair* and does not starve any particular type of traffic. For concreteness, we propose using a simple round-robin policy that rotates equally between all the buffers. Specifically, when u sees an $EMPTY$ slot in ring R_j, it does the following:

(1) Verifies that it has not transmitted more than $k-1$ packets on R_j in the last p clock ticks (i.e., it has access-control quota).
(2) Picks a packet from the next (round-robin) buffer among the single LQ_j buffer, the $d-1$ BQ_{ij} buffers, and the $(n-1)(d-1)$ RQ_{ji} buffers for which it has flow-control quota.

Remarks:

- The simple round-robin policy seemed to perform reasonably well in our simulation study (Section 6), however other policies are possible. Note that a bad choice of policy may easily lead to starvation: E.g., if $LOCAL$ traffic always has priority over bridged traffic, and there is a high $LOCAL$ traffic load, the quotas for some remote nodes will eventually decrease to 0. Investigating the effects of various policies is left for future work.
- The *VRing* protocols are deadlock-free since every packet that is placed in a slot will reach either its destination or its bridge node during the current cycle, and will be removed from the slot there (the bridge node is guaranteed to have a buffer for it). Thus progress can always be made and deadlock is impossible. However, if the packet selection policy at the bridge is not fair and leads to starvation, packets may remain in the access queues (outside the network) indefinitely.

5 Properties of the *VRing* Routing and Flow Control Protocols

In this section we analyze the properties of our protocols. We first show that our flow control protocol is congestion free. Then we prove that our protocols ensure a bounded network delay which is independent of the traffic pattern. Finally, we discuss the implications of the path redundancy inherent to GQ-based networks: excellent network survivability, and a simple path selection algorithm which allows easy load-balancing.

To show that the flow control is congestion free, we prove in Theorem 5.2 that no bridge buffer ever overflows. For the proof we first need a technical lemma.

Lemma 5.1 $|BQ_{m+1}| - |BQ_m| \leq B_m$.

PROOF. Packets sent under quota B_m will appear at u only *after* the quota matrix returns to u at the end of cycle m, thus they will be counted in u's cycle $m+1$. Hence the total number of arrivals at BQ_{ij} during cycle $m+1$ is at most B_m. Therefore even if no packet is forwarded onto R_j (i.e., no packet leaves BQ_{ij}) during cycle $m+1$, the change in the buffer's occupancy during this cycle is at most the capacity B_m advertised at the start of the previous cycle. □

Theorem 5.2 $|BQ_{m+1}| \leq MAXBQ$ *for all cycle* m.

PROOF. Assume that $B_0 = 0$. It is easy to see that $B_1 = MAXBQ$, so the theorem holds for $m = 0$. For $m \geq 2$ we apply Lemma 5.1 (twice) and rule (4) to obtain that

$$\begin{aligned} |BQ_{m+1}| &\leq |BQ_m| + B_m \\ &= |BQ_m| + (MAXBQ - |BQ_{m-1}| - B_{m-1}) \\ &= MAXBQ + (|BQ_m| - |BQ_{m-1}| - B_{m-1}) \\ &\leq MAXBQ + (B_{m-1} - B_{m-1}) \quad = \quad MAXBQ. \qquad \square \end{aligned}$$

We proved that there is no buffer overflow at the end of each cycle. However, it is easy to see that at intermediate time points within the cycle the buffer cannot overflow either, since the advertised capacity computed by rule (4) is conservative and does *not* assume that any packets leave the buffer. In fact even if no packets ever leave the buffer, it will not overflow—the flow control will quench all the sources sending via this buffer by setting their r^{ij} quotas to 0.

Remark: For simplicity in the proof we assumed that $B_0 = 0$. However, our simulation studies show that using this initial setting leads to a long period of oscillations in the quotas, in which nodes alternate between advertising a capacity $B_m = MAXBQ$ and a capacity $B_m = 0$. Our simulations show that the oscillation is eliminated and the quotas stabilize much faster if we set $B_0 \leftarrow MAXBQ/2$. It is easy to see that this is safe when the network performs a "cold start" and no packets are in flight.

Let $\Delta(uv)$ denote the network delay from node u to v. Recall that D is the time it takes for a slot to rotate all the way around a single ring in the network. The network delay for $LOCAL$ packets is clearly bounded by the propagation delay on a ring:

Fact 5.3 *If u and v are neighbors on some ring R_i then $\Delta(uv) \leq D$.*

When a bridge node x sees an $EMPTY$ slot on ring R_j, it needs to select which packet to put in it from the packets waiting in its LQ_j, RQ_{ji}, and BQ_{ij} buffers (recall Section 4.4). The bound on the network delay for $REMOTE$ packets depends on the guarantees that can be given for the packet selection policy used at node x (and on the rotation time D).

Proposition 5.

is subject to the respective $r_u^{ij}(x)$ quota, and for this u needs to know the identities of the bridge x and the ring R_j. Thus the *VRing* protocol uses a form of source routing.

However, our protocols can use the networks' path redundancy to balance the traffic load on the rings. This is done by a *path selection policy*: Since there are multiple disjoint paths between u and v (in fact there are d such paths in a GQ-based network of degree d), u can select any of these paths. We do not advocate any particular path selection policy in this paper. In the simulation studies (Section 6) we implemented a simple round-robin path selection policy.

The same path redundancy also implies desirable survivability properties for the network. It is easy to see that the network can survive $d-1$ link or node failures without becoming disconnected. For *REMOTE* packets, essentially the same routing protocol can be used even when some links or nodes fail.

LOCAL packets would require slightly different routing in failure mode. A *LOCAL* packet $P(v)$ may need to be sent to some other node z which shares a (still functioning) ring R_j with the destination v, and z would then forward the packet to v. Note that this bypass route traverses three rings and two bridges (one en-route to z and z itself). For the first part of the route the packet $P(v)$ would need to be converted to a special *REMOTE* packet.

6 The Simulation Study

In order to evaluate the performance of our protocols on networks that enjoy the one-bridge property, we conducted a simulation-based study. We first tested the behavior of our topology-tailored *VRing* protocols on their own. Then we compared their performance to that of the generic *INet* protocols. We refer the reader to [SWY99] for details of the *INet* protocols.

For this purpose a discrete event simulator program was written (in C++). We ran the simulator using a variety of networks and under many traffic load conditions.

6.1 Description of the Simulation Model

The basic unit of time in the simulation is a clock tick. In the simulator's basic mode of operation, every clock tick, each node in the network generates new traffic, and then for every ring the node is a member of, it receives the content of a slot, processes it, and possibly selects and sends a packet using the same slot. At the end of the clock tick the slots on all the rings rotate one position clockwise.

In the networks we implemented the nodes and rings are all equivalent to each other, and the nodes are symmetrically arranged on each ring. In particular the number of inter-node clock ticks between any two consecutive nodes on a ring is the same, and is a simulator parameter denoted by it.

When simulating the *VRing*, the basic mode of operation represents the network's *base bandwidth*. We are also able to simulate higher bandwidths, as long as they are integer multiples of the base bandwidth. This is implemented by using an integer *bandwidth factor* (bw). So in fact during each clock tick, each node processes bw slots, and $bw = 1$ indicates the base bandwidth with one slot per clock tick. The total number of slots on a ring of size n is $p = n \times it \times bw$. The access control quota that each node has is therefore $k = \lfloor p/(n-1) \rfloor = \lfloor it \times bw \times n/(n-1) \rfloor$.

In *VRing*, A node's packet selection policy, for each ring R_j it belongs to, is a round-robin policy. The policy rotates equally among the single LQ_j buffer, the $(n-1)(d-1)$ RQ_{ji} buffers and the $d-1$ BQ_{ij} buffers.

In *INet*, a node has two unbounded access buffers (queues) for each link it is adjacent to, which we denote by AQ and ARQ. The AQ buffers hold new packets, while the ARQ buffers hold packets that need to be retransmitted. In addition, a node has one bounded BQ buffer per link. These BQ buffers model the layer 3 queues. Congestion occurs when a BQ buffer is full. Packets move into a slot from the outgoing link's BQ buffer. The packet selection policy sets the priorities of packets entering a BQ buffer: First are packets arriving on (other) incoming links; next are retransmitted packets, from that link's ARQ buffer; and last are new packets, from the link's AQ buffer.

An important parameter to the simulator is the maximal size of the BQ buffers ($MAXBQ$). The other buffers (LQ, RQ, AQ, ARQ) were implemented as unbounded.

6.2 The Traffic Model

To simulate bursty data traffic we use an "on/off" traffic model, which is controlled by two parameters: the burst probability P_b and the maximum burst length M_b. Every clock tick each node u flips an independent probability-P_b coin for each other node v. If the coin shows "heads" u picks a value $1 \leq \ell \leq M_b$ uniformly at random, and generates a burst of ℓ packets with destination v. Thus the average number of packets that a node injects into the network each clock tick is $ppn = N \times P_b \times M_b/2$. In our graphs we typically use ppn ("packets per node") as our x-axis, since this quantity is easily comparable to a node's degree d: In *VRing*, a node can transmit at most d packets every clock tick, one on each ring it belongs to. In *INet*, a it can transmit at most $2d$ packets every 2 clock ticks.

In the *INet* there is a single path for each source-destination pair. However, in the *VRing* there are multiple paths available to the routing, so we need to specify our path selection policy. In our implementation of *VRing*, path selection is done burst-by-burst, so all the packets that belong to a single burst follow the same path through the network. If the source u and destination v are neighbors on some ring R_i ($LOCAL$ bursts) then the whole burst is placed in u's LQ_i buffer. As for $REMOTE$ bursts, recall that there are d disjoint one-bridge paths between u and v in a GQ-based network, each corresponding to a different RQ_{ij} buffer. To balance the load on the different rings, we again use a round-robin policy, this time for path selection: u cycles among the d possible paths it has to v, burst by burst.

	Load/BW	**Buffer Size**	***INet* Comparison**
Traffic parameters			
Max. burst size (M_b)	20		
Burst probability (P_b)	* .0002–.015	* .0007, .0010, .0023	.0002–.0080
Packets per node (ppn)	.0924–6.93	.3234, .4620, 1.0626	.0924–3.6960
Network parameters			
Number of nodes (N)	45		
Ring size (n)	5		
Degree (d)	3		
Inter-node ticks (it)	20		
Bandwidth factor (bw)	* 1, 2	1	1
Slots per ring (p)	* 100, 200	100	100
Access control quota (k)	* 25, 50	25	N/A
Buffer Management			
Packet selection policy	round-robin		$BQ \gg ARQ \gg AQ$
$MAXBQ$	20	* 3–100	20

Table 3
Simulation model parameters for the reported studies. Value ranges prefixed by a "$*$" indicate parameters that were varied in each study.

6.3 The Performance Measures

We gathered statistics on the following types of performance measures. For each of them we calculated the mean, range, and standard deviation. The curves in the sequel all show the mean values.

The **network utilization** is the fraction of slots containing a packet out of the total number of slots. In the *VRing*, each packet is transmitted exactly once since there are no retransmissions, and hence the utilization is well defined. However, in the *INet* a packet may be transmitted several times before it reaches its destination successfully. Thus, we need to separate the raw link utilization from the effective utilization, which only measures the fraction of slots carrying "useful" packets. In other words, the effective utilization is the raw utilization normalized to capture only the successful transmissions. For this purpose, we also compute the **throughput**, which is the ratio between the number of packets successfully delivered to their destinations, and the total number of packets sent (new and retransmitted). The effective utilization is defined as the raw link utilization multiplied by the throughput of the *INet*.

The **access delay** of a packet is the number of clock ticks from the moment the packet is placed in the access buffer (LQ or RQ in the *VRing*, AQ in the *INet*) until it is placed in the network.

The **network delay** of a packet is the number of clock ticks from the moment a packet is placed in a slot at its source until it reaches its destination. This includes the waiting time in the network buffers. In the *INet* the network delay also includes retransmissions, with possible waits in ARQ buffers.

Each point in our graphs represents the output of a run of 12,000 clock ticks. Transients [Jai91] were eliminated by discarding the first 2000 clock ticks. For such runs, the graphical height of the 95%-confidence intervals [9] was very close to the size of the symbols depicted on the curves. Therefore in order to reduce the complexity of our graphs, we do not show these confidence intervals.

---

[9] A 95% confidence interval means that that value appears within the specified interval with probability 0.95. See [Jai91] for a precise definition of confidence interval.

6.4 Results and Interpretations

6.4.1 The Load/BW Study

In this study we show the *VRing*'s performance under increasing traffic loads, for two bandwidths. We performed the same study using several different networks, all of which gave qualitatively similar results. Here we report the results obtained on a network based on the so called $Q(5, 2)$-dual GQ (see Table 2). The parameters we used are listed in Table 3.

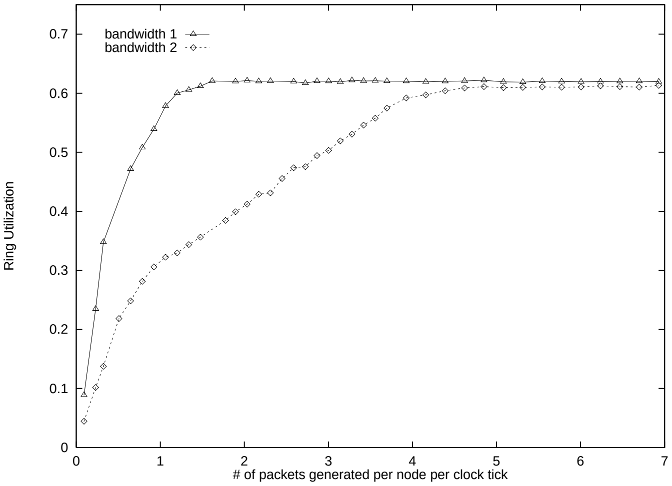

Fig. 2. Link utilization, Load/BW study

We see in Figure 2 that as the traffic load increases the link utilization rapidly grows up to a "knee point", where the utilization levels off at ≈62.5%. This fits the prediction of Corollary 3.7 very well: The packet destinations are chosen uniformly in our traffic model, so $E[PLen] = p/2$. And since the ring size is $n = 5$, the bound is $1/2 \times 5/4 = 0.625$. This behavior indicates that the *VRing*'s utilization bottleneck is the access control rather than the flow control protocol, since Corollary 3.7 is a consequence of the access control.

When the bandwidth increases, the network reaches its maximal utilization at higher traffic loads: the knee point is $ppn = 1.4784$ and 4.3890 for bandwidth factors $bw = 1$ and 2, respectively. We observe that the network utilization increases more slowly than the bandwidth: Figure 2 shows the knee point for $bw = 2$ is at a traffic load approximately *three* times higher than that of $bw = 1$, yet the $bw = 2$ provides twice the base bandwidth.

The waiting times for packets in the LQ and RQ buffers are shown in Figure 3. Since the protocols are congestion-free, when the network becomes saturated the access control causes the buffers to explode and the waiting time increases rapidly. Our results show that the waiting time in the RQ buffers is smaller, i.e., the RQ buffers can sustain a higher traffic load before exploding. This

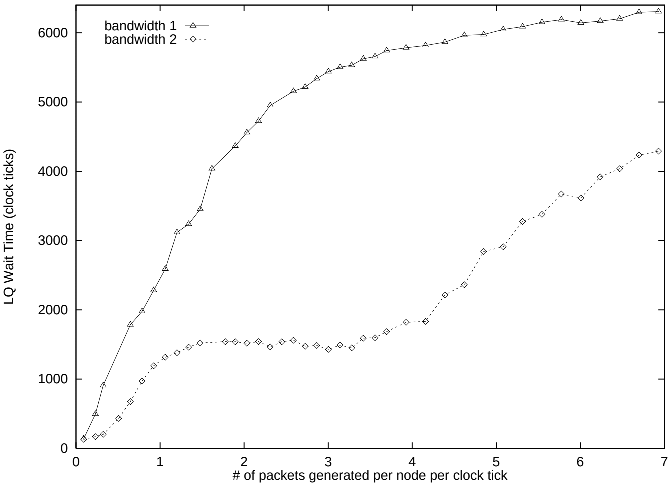

Fig. 3. Waiting time for *LOCAL* packets in the *LQ* buffers, Load/BW study.

is somewhat surprising at first glance since most of a node's destinations are *REMOTE* (in the $Q(5,2)$-dual network a node has 12 neighbors and 32 non-neighbors). But this is mitigated by the fact that a node has *more RQ* buffers. The LQ_i buffer sees traffic destined to the $n-1$ nodes on R_i. In comparison, packets placed in RQ_{ij} are destined to one of the $n-1$ nodes on R_j—but there are d disjoint paths to each of them, each with a *different RQ* buffer at the source. Since we use round-robin path selection, any specific RQ_{ij} buffer thus sees only $1/d$ of the traffic that an LQ buffer sees.

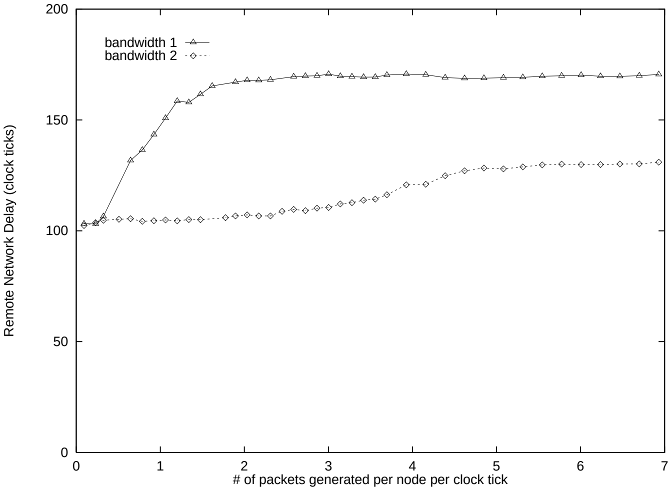

Fig. 4. Network delay for *REMOTE* packets, Load/BW study.

In Figure 4 we see the average network delay for *REMOTE* packets. This delay consists of the propagation delay on each of the two rings and the waiting time in the bridge's BQ_{ij} buffer. The propagation delays contribute $2 \times E[PLen]$ clock ticks (100 ticks in this case), and any additional delay is due to the wait in a BQ buffer. Note that the average delay is much less than the bound of Proposition 5.4: In the $Q(5,2)$-dual network the round-robin packet selection guarantees that $b = k/(\#buffers) = 25/11$ packets will be removed from each

BQ buffer in each cycle, so Proposition 5.4 gives a bound of 1080 ticks, whereas the maximal average delay we observe is 170.5 ticks (for $bw = 1$). We note that the network delay for the $REMOTE$ packet remains stable regardless of the bandwidth.

$LOCAL$ packets are delayed only $E[PLen] = 50$ ticks regardless the of traffic load since they do not pass a bridge and so they do not suffer from any waiting time (graph omitted).

6.4.2 The Buffer Size Study

In this study we tested the effect of the size of the BQ buffers on the network performance, again using the $Q(5, 2)$-dual network. Table 3 gives the parameter settings we used.

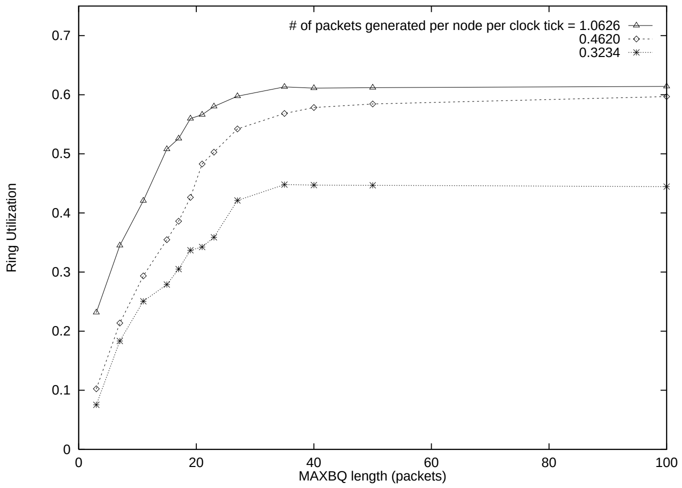

Fig. 5. Link utilization, Buffer Size study

Figure 5 shows that the link utilization is insensitive to the actual $MAXBQ$ value—as long as it is not too small. When $MAXBQ$ is too small the flow control protocol denies access to $REMOTE$ packets, thus nodes are unable to use all of their access control quota. This causes the rings to be under-utilized, and the $REMOTE$ delay to be very high. When $MAXBQ$ is extremely low (below 4 in this case) the data shows that the RQ buffers become unstable. However, larger buffers cause increased network delay for remote packets. It seems that if $MAXBQ \geq (n - 1) \times M_b/2 = 40$, i.e., allowing each neighboring node on the source ring to generate one average-sized burst per cycle, then the maximal theoretical link utilization is reached for $ppn \approx 1$. For the other studies we tried we used a lower value of $MAXBQ = 20$, which gives better end-to-end delays, but the link utilization is maximized only at higher traffic loads.

6.5 The INet Comparison

After evaluating the performance of the *VRing* protocols on their own, we turn to a comparison with the generic *INet* protocols.

6.5.1 Comparison of Network Utilization

In Figure 6 we show three curves. The *VRing utilization* is simply the ratio of occupied slots vs the total number of slots in the ring links. We see in Figure 6 that as the traffic load increases the *VRing* utilization grows roughly linearly up to a "knee point", where the utilization levels off at ≈62%.

The *INet* raw utilization curve, is the raw link utilization, computed similarly to the *VRing* utilization. We observe that the *INet* raw link utilization grows up to about 90%.

However, this is misleading. Although the links in the *INet* are occupied up to ≈ 90%, many of the occupied slots are carrying retransmission packets. The *INet effective utilization* curve shows that the effective utilization levels off at ≈ 35%, which is close to the reported throughput of TCP traffic over the Internet [TMW97].

To conclude, the *VRing* provides an effective network utilization which is almost twice as high as that of the *INet*, since it has no loss due to congestion, and no retransmissions.

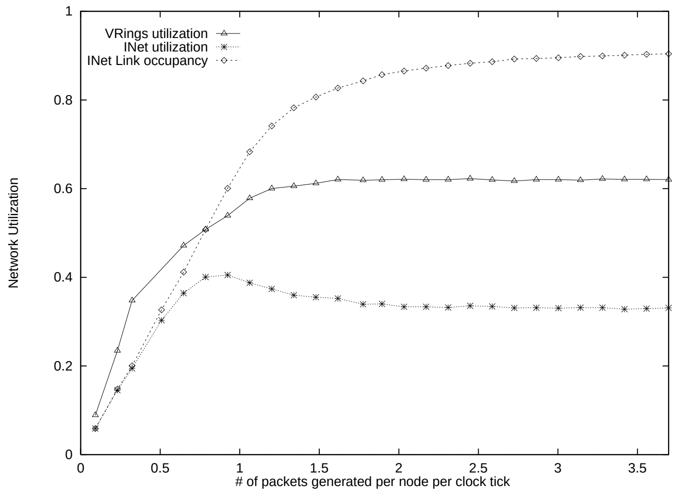

Fig. 6. Link utilization, as a function of the traffic intensity.

6.5.2 Delay Comparison

We break the delay into two components, which are the access delay, and the network delay (i.e., the end-to-end propagation and queuing delay). Figures 7–8 show the behavior of the delay a function of traffic intensity.

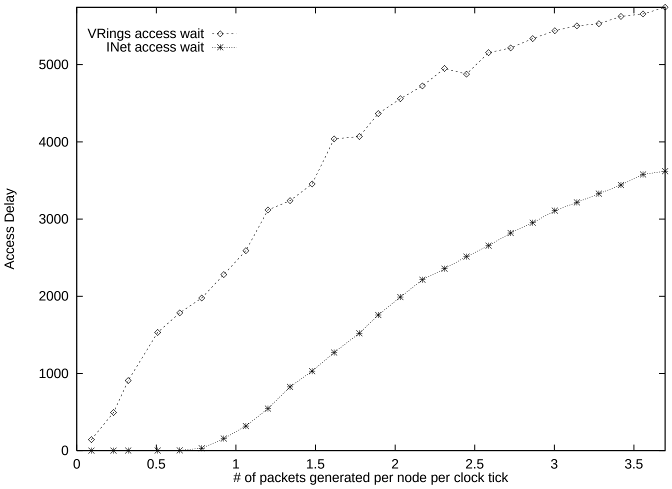

Fig. 7. Access delay, as a function of the traffic intensity.

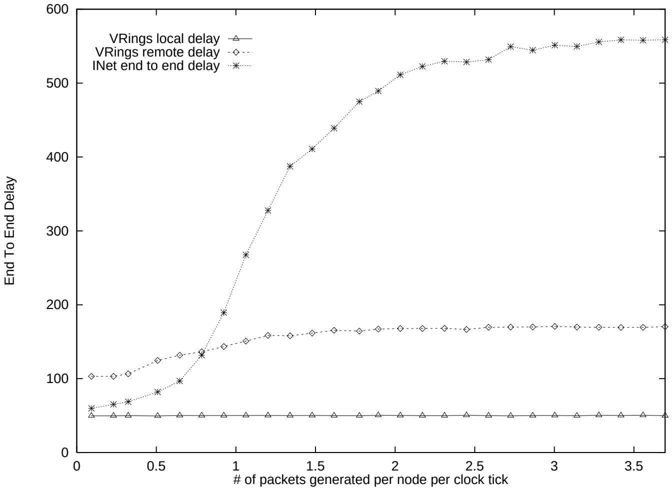

Fig. 8. End-to-end delay, as a function of the traffic intensity.

Figure 7 shows that the access delay is approximately twice higher in the *VRing* than in the *INet*. This is expected since in the *VRing* the network accepts only what it can deliver with no loss due to congestion. Thus, packets have longer waiting time at the network boundary.

Figure 8 shows that once packets enter the *VRing*, the end-to-end network delay is significantly lower in the *VRing* than in the *INet*. *LOCAL VRing* traffic has a very short delays (≈ 50 clock ticks), which is independent of the traffic intensity. The delay of *REMOTE VRing*traffic increases slowly with the traffic intensity, but is bounded by ≈ 170 clock ticks. In contrast, the *INet* network delay grows rapidly with traffic intensity. This is since more traffic causes more congestion, which in turn cause retransmissions.

Another aspect of the network delay is its variability. Applications such as audio and video are relatively tolerant of network delays, but are severely degraded by high delay variability and jitter. To quantify the variability, we computed the standard deviation of the network delay. Our results show that in the *VRing*, the variability of the network delay is quite small and remains stable even the traffic intensity increases. However, the *INet* delay variability is large, and increases with the traffic intensity. For high traffic intensity, the *INet* delay is ≈ 550 with a standard deviation of ≈ 800, which implies that a significant fraction (say 15%) of packets are delayed more than 1350 clock ticks. Such a variability of the delay implies that the jitter is also very large.

However, we note that the total delay (i.e., sum of access delay and the network delay) is still significantly lower in *INet* than *VRing*.

7 Conclusions

We have described a new methodology for the design of multi-ring networks, which is based on combinatorial design theory. Our multi-ring networks have the one-bridge property: the path between any two nodes is either confined to a single ring, or traverses exactly two rings (passing through a single bridge node). Such networks have been constructed using combinatorial block designs called Generalized Quadrangles.

We presented two alternative types of routing and flow control protocols that may be used on networks with the one-bridge property: topology-aware (*VRing*), and topology-indifferent (*INet*). The *VRing* approach provides congestion-free and fair routing with bounded network delays, and depends on the one-bridge property. The *INet* uses generic, Internet-like protocols with shortest path routing. We examined the performance of these approaches with an extensive simulation study.

Our results show that the total delay in the *VRing* is higher than the *INet*. However, the *VRing* provides lower network end-to-end delay with lower variability and jitter. Moreover, the effective utilization in the *VRing* is substantially higher, due to the retransmissions in the *INet*. We conclude that from the network provider's perspective, the *VRing* provides much better network utilization, and allows the provider to give the network's users QoS guarantees regarding available bandwidth, fairness, and jitter. From a user's perspective, the *INet* may be a better choice if the total delay (the sum of access delay and the network delay) is the most important network property. However, for user applications that are more sensitive to jitter than to delay, such as audio and video transmissions, the *VRing* approach may be advantageous.

References

[ANS88] ANSI T1.105-1988. American National Standard for Telecommunications, Digital Hierarchy—Optical Interface Rates and Format Specifications, 1988.

[BCK+83] W. Bux, F. H. Closs, K. Kummerle, H. J. Keller, and H. R. Mueller. Architecture and design of a reliable token-ring network. *IEEE Journal of Selected Areas in Communications*, SAC-1(5):756–765, November 1983.

[BOY97] M. Baldi, Y. Ofek, and B. Yener. Adaptive real time group multicast. In *IEEE INFOCOM'97*, 1997.

[CD96] C. J. Colbourn and J. H. Dinitz. *The CRC Handbook of Combinatorial Designs.* CRC Press, Boca Raton, 1996.

[CHG+01] N. Cole, J. Hawkins, M. Green, R. Charma, and K. Vasni. Resilient packet rings for metro networks, August 2001. `http://www.rpralliance.org/articles/RPR_AllianceGOC.pdf`.

[CO93] I. Cidon and Y. Ofek. MetaRing - a full-duplex ring with fairness and spatial reuse. *IEEE Trans. Communications*, COM-41(1):110–120, January 1993.

[Dan98] P. H. Dana. Global positioning system overview. The Geographer's Craft Project, Department of Geography, The University of Texas at Austin, 1998. `http://www.utexas.edu/depts/grg/gcraft/notes/gps/gps.html`.

[DH98] B. Doshi and P. Harshavardhana. Broadband network infrastructure of the future: Roles of network design tools in technology deployment strategies. *IEEE Communications Magazine*, 36(5):60–71, 1998.

[DP91] A. De and N. Prithviraj. Ring-connected-ring (RCR) topology for high-speed networking: Analysis and implementation. *ACM SIGCOMM Computer Communications Review*, 21(3):33–44, July 1991.

[FN69] W. D. Farmer and E. E. Newhall. An experimental distributed switching system to handle bursty computer traffic. *Proc. ACM Symp. on Problems in Optimization of Data Communication Systems*, pages 1–33, 1969.

[Jai91] R. Jain. *The Art of Computer Systems Performance Analysis.* John Wiley & Sons, 1991.

[Kap96] E. D. Kaplan. *Understanding GPS, Principles and Applications.* Artech House, Boston, 1996.

[Kes97] S. Keshav. *An Engineering Approach to Computer Networking.* Addison-Wesley, 1997.

[LOSS94] C-S. Li, Y. Ofek, A. Segall, and K. Sohraby. Pseudo-isochronous cell switching in ATM networks. In *IEEE INFOCOM'94*, pages 428–437, 1994.

[LOY96] C-S. Li, Y. Ofek, and M. Yung. "Time-driven priority" flow control for real-time heterogeneous internetworking. In *IEEE INFOCOM'96*, 1996.

[LSBW98] E. Livermore, R. P. Skillen, M. Beshai, and M. Wernik. Architecture and control of an adaptive high-capacity flat network. *IEEE Communications Magazine*, 36(5):106–112, 1998.

[MB76] R. M. Metcalfe and D. R. Boggs. Ethernet: Distributed packet switching for local computer networks. *Communications of the ACM*, 19(7):395–404, July 1976.

[Ofe94] Y. Ofek. Overview of the MetaRing architecture. *Computer Networks and ISDN Systems*, 6–8:817–830, 1994.

[OY90] Y. Ofek and M. Yung. Principles for high speed network control: Losslessness and deadlock-freeness, self-routing and a single buffer per link. In *9'th ACM Symp. Princip. of Dist. Comp. (PODC)*, pages 161–175, August 1990.

[OY95] Y. Ofek and M. Yung. METANET: Principles of an arbitrary topology LAN. *IEEE Trans. on Networking*, 3(2):169–180, April 1995.

[PT84] S. E. Payne and J. A. Thas. *Finite Generalized Quadrangles.* Research Notes in Mathematics 110. Pitman, London, 1984.

[RGA85] C. S. Raghavendra, M. Gerla, and A. Avižienis. Reliable loop topologies for large local computer networks. *IEEE Trans. Computing*, C-34, No. 1:46–55, January 1985.

[RPR01] An introduction to resilient packet ring technology, October 2001. `http://www.rpralliance.org/articles/Whitepaper10-01.pdf`.

[RS86] C. S. Raghavendra and J. A. Silvester. A survey of multi-connected loop topologies for local computer networks. *Computer Networks and ISDN Systems*, 11:29–42, 1986.

[SWY99] Y. Song, A. Wool, and B. Yener. The performance of routing and control protocols on virtual rings. In *IEEE Global Communication Conf. – GLOBECOM'99*, pages 603–610, Rio de Janeiro, Brazil, December 1999.

[Tan96] A. S. Tanenbaum. *Computer Networks.* Prentice Hall PTR, 3rd edition, 1996.

[TMW97] K. Thompson, G. J. Miller, and R. Wilder. Wide-area internet traffic patterns and characteristics. *IEEE Network*, November 1997.

[Var86] A. Varma. Combinatorial design of bus-based interconnection structures. *IBM Research Report no: RC 12550*, 1986.

[WY99] A. Wool and B. Yener. Combinatorial design of multi-ring networks with combined routing and flow control. Technical Report 99-04, DIMACS, 1999. Available from `http://dimacs.rutgers.edu/TechnicalReports/abstracts/1999/99-04.html`.

[YOY94] B. Yener, Y. Ofek, and M. Yung. Topological design of loss-free switch-based LANs. In *IEEE INFOCOM'94*, 1994.

[YOY97] B. Yener, Y. Ofek, and M. Yung. Combinatorial design of congestion-free networks. *ACM/IEEE Transactions on Networking*, 5(6):989–1000, December 1997.