Dynamic Time Multiplexed Virtual Channels, a Performance Scalable Approach in Network-On-Chip Routers to Reduce Packet Starvation

Bharath Sudev and Leandro Soares Indrusiak

The Department of Computer Science, The University of York, United Kingdom YO10 5GH
{bs638, leandro.indrusiak}@york.ac.uk

Abstract. Latency performance is enhanced in Network-on-Chips using a variety of techniques, among which virtual channel approach is one of the most widely used. With virtual channels, high priority packets can pre-empt low priority packets but this can cause starvation of low priority packets which can be un-necessary in many situations. With this paper we present an improvement on the classical virtual channel approach which would provide a performance scalability feature that can reduce low priority packet starvation. This new system would allow the router to work in several performance levels so that if high priority packets are received ahead of their deadlines, the performance level of the router can be tweaked so that it would allow lower priority communication interlaced with the high priority communication.

Keywords: Dynamic Time Multiplexed Virtual Channels, Performance Scalability, Packet Starvation Reduction

1 Introduction

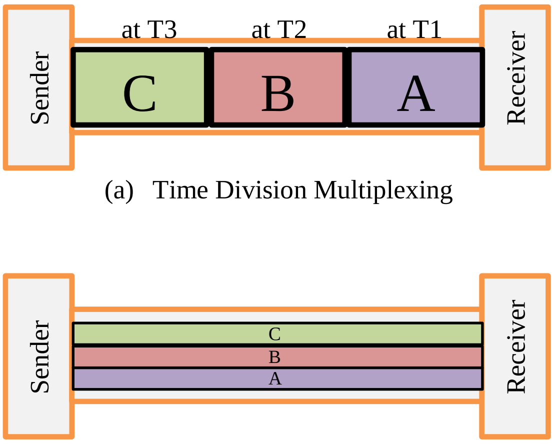
With Network-on-Chip (NoC) systems having packets flows with different priority levels, it is important to have hardware to ensure packet predictability based on priority level. With non-preemptive NoC designs, this can be difficult as tailbacking and Head of line blocking of packets can result in undesired magnitudes of latency, even for the highest priority packets. One of the most effective ways to ensure prioritised service is the use of virtual channels where the routers would be able to pre-empt packet flows so that higher priority ones can be transmitted as soon as possible. This is done by introducing multiple service levels for packets so that a higher priority service level packet would be able to pre-empt any lower priority service level communication.

In this paper we present an approach by which such virtual channel based systems can be improved so that starvation of low priority packets can be decreased when possible by interlacing low priority communication with high priority flows.

2 Background

Packet Latency pefomance can be improved in NoC systems using a varierty of techniques and Time Division Multiplexing (TDM) is one of the important ones in this context. TDM is widely used in NoC archtectures (like in Aethereal [1] and Nostrum [2]) where each router would have a slot table and the functionality of every router at any point of time would depend on the entry on the respective slot table. TDM provides packet delivery with zero variation in latency but the functionality of the router would be static. For that reason, to enable any new packet flows the whole set of slot tables would have to be re-configured which is complicated and time consuming thus limiting scalablity.

Link Division Multiplexing (LDM) introduced by Morgenshtein et al in [3] presents a different approach to improve packet latency predictability. LDM aimed at improving predictability by dividing the communication link into multiple sections so that multiple packets flows would be able to utilise the same link simultaneously. **Fig. 1** shows a comparison between TDM based transmission and LDM based transmission and it shows how packet flows A, B and C are treated in both cases.



(a) Time Division Multiplexing

(b) Link Division Multiplexing

Fig. 1. TDM vs LDM (a) TDM based Transmission (b) LDM based Transmission

Fig. 1(a) show the functionality of a TDM based transmission and it can be seen that at each time instant a specific packet is given the full bandwidth of the communication link. For example, if we assume that a flit is transmitted in every clock cycle, on the first clock cycle (T1), packet A is transmitted using the entire bandwidth. At the next clock cycle (T2) B gets transmitted and at the third clock cycle (T3) C gets transmitted.

With LDM however, the whole bandwidth is shared between the three packets A, B and C as shown in **Fig. 1**(b).Even though multiple packets would be able to utilise the same link, priority rules can be applied to the transmission by allocating more bus lines (bandwidth) to qualifying packets.

One of the most widely use predictability enhancement techniques is the Virtual Channels [4] approach (like in Nostrum [2] and MANGO [5]). It is an extension to Wormhole flow control aimed at reducing deadlocks, improving network utilisation and for improving performance under congestion. By the use of Virtual Channels, the bandwidth of a physical connection path is multiplexed into separate logical channels so that multiple flits can share the same path.

Introduced by Dally in [19], the virtual channel technique relies on the use of multiple buffers for each channel on the network so that communication through a link is possible even if a flit is blocked on a link.

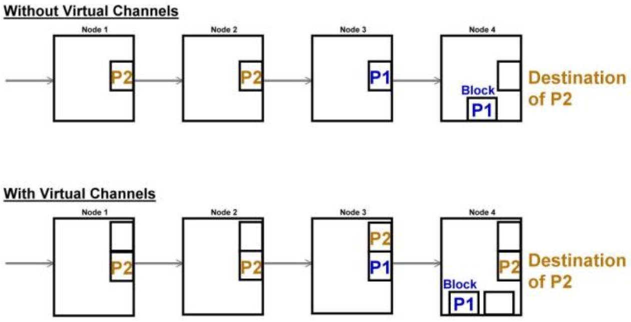


Fig. 2. Virtual Channel Functionality

Fig. 2 depicts a comparison between routers without virtual channels and routers with virtual channels. Without virtual channels P2 flits are stuck behind P1 flits when P1 is blocked, as P1 and P2 share ‘node 4’ for transmission. With the use of virtual channels, it can be seen that P2 packets would be able to reach the destination even if the shared node has blocked packets as virtual channels use multiple input buffers for each channel. Mello et al in [6] compared performance of a Hermes [7] NoC with and without virtual channels *(by varying the number of virtual channels from 1)* and the test reveals reduction of average latency of more than 50% for their 8x8 NoC with uniform load. As virtual channel routers require separate buffers and arbitration/control logic for each service level, virtual channel based routers have high hardware overheads.

There were also predictability enhancement techniques based of adaptive approaches like [8], [9] and [10] where they used dynamically adaptive routing by monitoring traffic in the NoC in real-time. While Ge et al. in [8] utilised a centralised monitoring module to alter the source routing depending on the traffic on the NoC, Cidon et al. in [9] utilised traffic maps in their design for a similar mode of operation. Rantala et al. in [10] dealt with adaptability in a distributed perspective where the source routing at each network interface was altered depending on the congestion information retrieved from neighbouring routers.

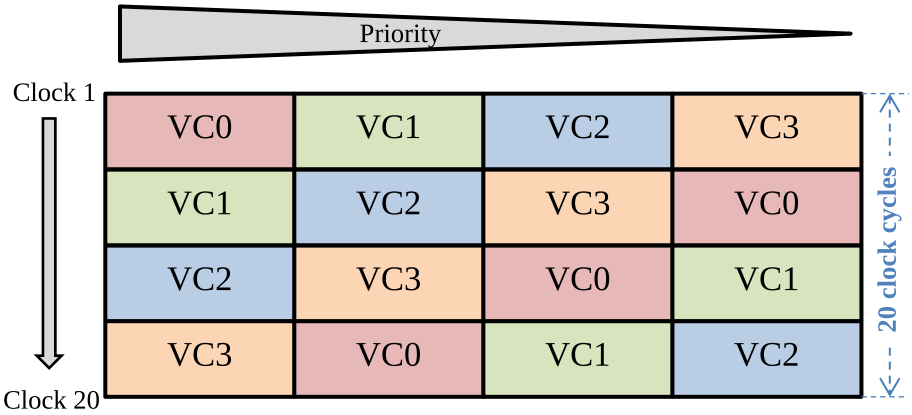
With our previous work in [11], [12] and [13], we tried to improve packet predictability by resolving Head of line blocking and tailbacking situations associated which non-preemptive NoC packets. We used adaptive routers combined with adaptable packets so that Head of line blocking would be resolved by forwarding the priority of the blocked packet to the blocking packet. Tailbacks were resolved by splitting all low priority communication which blocks higher priority communications.

With Dynamic Time Multiplexed Virtual Channels (DTMVC), we are trying to improve upon the virtual channel approach by enabling the router to take account of the packet latency performance so that its performance can be scaled. This would reduce starvation of low priority packets as seen with the classical virtual channel approach. DTMVC can be seen as a dynamic hybrid between TDM and virtual channel approach.

3 Dynamic Time Multiplexed Virtual Channels

With basic virtual channel design, the highest priority active virtual channel gets the entire bandwidth available which can cause starvation of lower priority virtual channel packets. Unless there are late packets of higher priority virtual channels, this kind of extreme approach brings about unnecessary starvation which can be averted. DTMVC provides an opportunity to scale the performance enhancement level of the router so that unnecessary starvation can be avoided. In this paper, we depict the functionality of a router that supports four performance settings.

With DTMVC, the operating time of the design is divided into recurring timeframes. For example, consider a time frame of size 20 clock cycles shown in **Fig. 3**.



VC0	VC1	VC2	VC3
VC1	VC2	VC3	VC0
VC2	VC3	VC0	VC1
VC3	VC0	VC1	VC2

Fig. 3. Example time frame at PS3

Here we assume that there are four virtual channels; VC0, VC1, VC2 and VC3. The example depicts the lowest of performance level; Performance Setting 3 (PS3) where the time frame is divided into equal time slots for each virtual channel. We can see that in the first quarter of the time frame, Virtual Channel 0 would have the highest priority followed by VC1, VC2 and VC3. During the second quarter, VC1 would get the highest priority followed by VC2, VC3 and VC0. Likewise with the next two quarters, the rest of the two virtual channels would get equal time slots as the highest priority virtual channel. If the system detects that the latency performance of VC0 is poor resulting in late packets, the router has in built logic to adjust the time frame so that VC0 gets more time slots as the highest priority virtual channel at the cost the slot times allocated to the other channels. So, the router would then switch the Performance Setting to 2 (PS2) thereby modifying the time slot allocation. A possible slot allocation for PS2 is shown in **Fig. 4**.

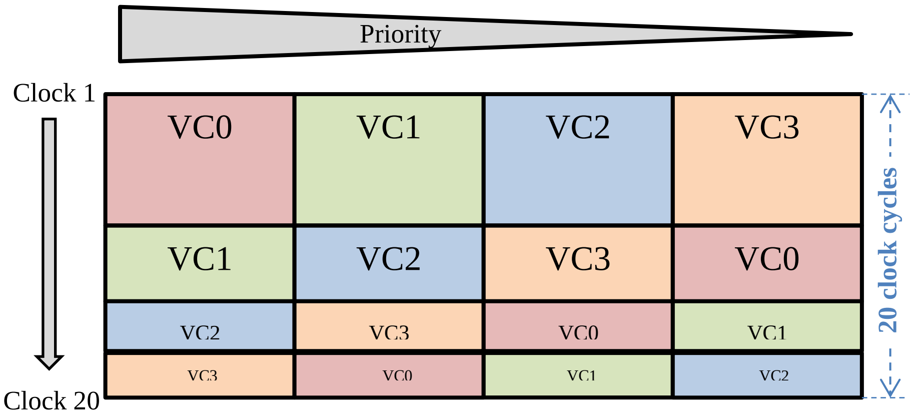


Fig. 4. Example time frame at PS2

Here you can see how VC0 spends more time as the highest priority virtual channel than the previous setting. With subsequent two PS levels, this extension of VC0's slot time would increase ultimately getting to PS0 where VC0 would always remain as the highest priority channel and the performance would then be like a classical virtual channel design.

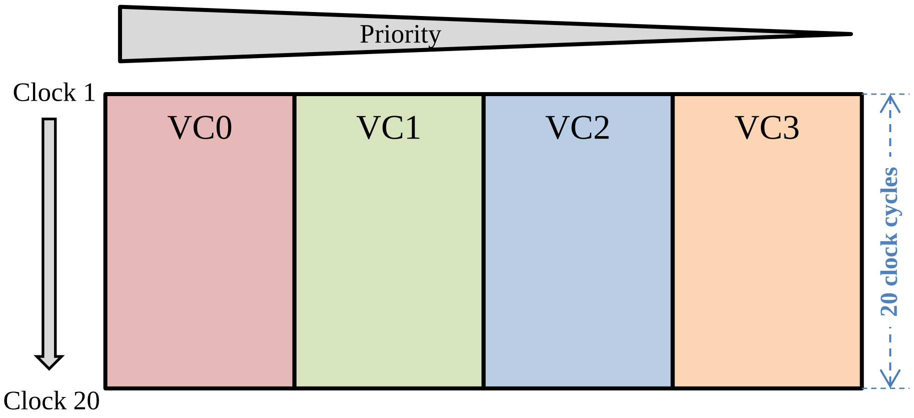


Fig. 5. Example time frame at PS0

The instruction to change performance setting can be provided by a centralised monitor or the packet receptions when late packets are encountered.

3.1 Prototype Architecture

The prototype router was designed as a five port architecture based roughly around Hermes [7] hence employing XY-routing and wormhole switching to reduce hardware requirements. The design uses a uniform mesh topology and unlike Hermes, each packet header includes a priority value which is used by the arbitration unit of the router to resolve contention between packets over output ports. As shown in **Fig. 6**, the routers have buffered input ports which on reception of a packet header employ XY-routing to set the 'port request' register and the 'priority' register in accordance with the destination and priority information carried.

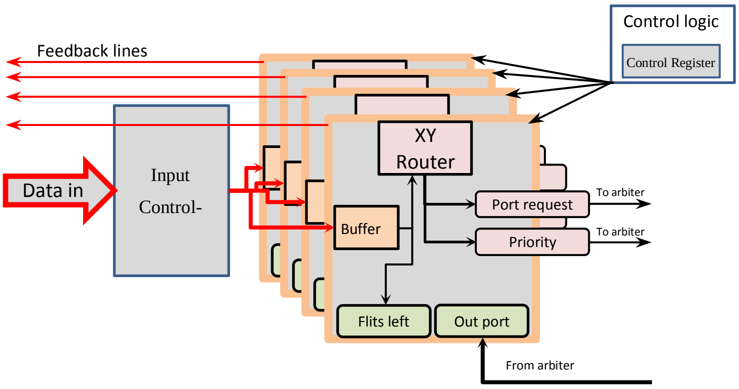


Fig. 6. Input port architecture

The arbitration unit in the router then checks 'port request' and 'priority' registers of all input ports to provide arbitration to the qualified ports. The arbiter then establishes connection by setting the 'out port' and 'flits left' registers on the input port. This permits the input port to send flits to the allocated output port so that the flits could be transferred away through the communication links. As the flits are being transferred, the input port also decrements the value in the 'flits left' register so that when the value reaches zero, the connection can be closed by re-setting the 'out port' register value to zero.

For each of the virtual channels supported, this infrastructure is duplicated and there is a 'control register' at each input port which designates the active input virtual channel at that moment. To prevent blocking of paths, each input port also has connection lines (for each service level) to the output port of the nearby router to notify the sender of the un-availability of buffer space in the receiver and hence to stop communication. The control logic evaluates the active time frame and other competing flows to enable appropriate input port for data transmission. The priority assignment for the virtual channels is accessed from the slot table so that the appropriate input port can be enabled. The control logic employs a counter to select appropriate priority assignments at that point of time and an example slot table is provided as **Table 1**. The router has internal logic to alter the priority assignments inside the slot table so that the performance setting can be altered dynamically.

Table 1. Slot table inside input port at PC3

Counter value	Priority of VC0	Priority of VC1	Priority of VC2	Priority of VC3
0	0	1	2	3
1	0	1	2	3
2	3	0	1	2
3	3	0	1	2
4	2	3	0	1
5	2	3	0	1
6	1	2	3	0
7	1	2	3	0

3.2 Evaluation

Consider a four virtual channel NoC with four linearly distributed perfomance settings.

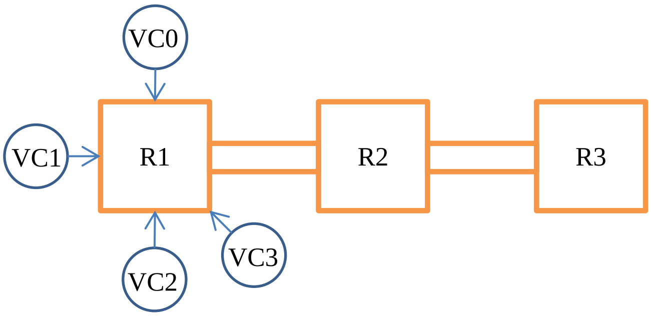


Fig. 7. Performance estimation example

To estimate the effect on the latency with the perfomance settings PS3, PS2, PS1 and PS0, consider the example in **Fig. 7**. We assume that there are packets of virtual channel 0, 1, 2 and 3 from the north, west, south and local ports of router R1 destined to router R3. Assuming a packet width of 100 and a packet generation period of zero, the latency perfomance of the packets can be seen in **Fig. 8**.

In **Fig. 8**, it can be seen that at PS0, the whole bandwdith is given to virtual channel zero and hence the latency is seen to be the lowest but this results in starvation of the other service level packets. This is similar to what we see with the classical virtual channel approach. With the perfomance setting set to one, virtual channel 0 suffers a minor increase in latency as a result of the router allocating time slots to the other service levels. With this setting, packets from virtual channels 1, 2 and 3 gets transmitted but at very high latencies. Similar effect is seen with PS2 and, with PS3 all the virtual channels are provided equal time slots resusting in equal latency perfomance.

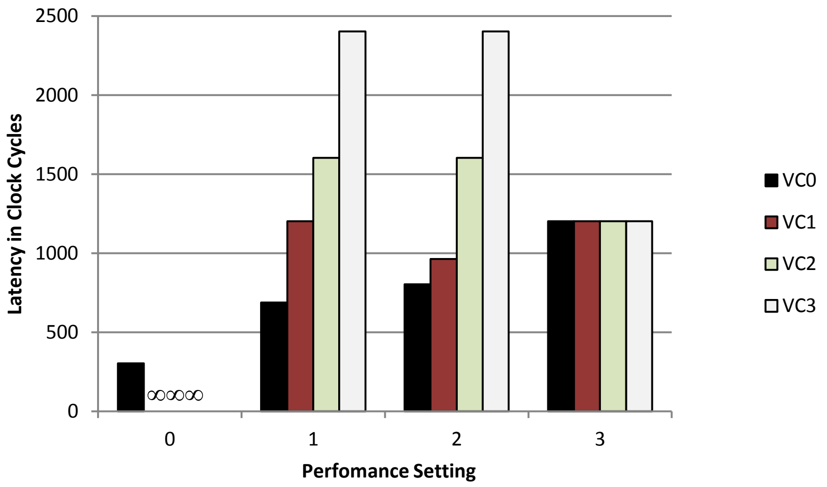


Fig. 8. Maximum no contention latency

The elementary prototype was designed in Bluespec System Verilog and in **Fig. 9**; the hardware comparison with our virtual channel based baseline can be seen.

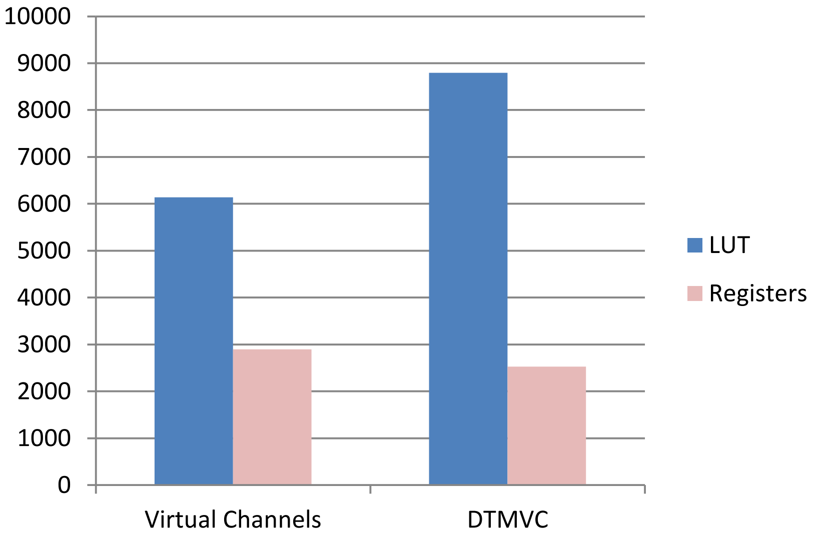


Fig. 9. Hardware overhead comparison

The hardware overhead with DTMVC was found to be 40% lookup tables and 14% registers more than the classical virtual channel based NoC. The current prototype has limited functionality as the logic does not account for virtual channels without any transactions, so that a lower priority virtual channel can be provided access even if the slot table depicts the higher priority virtual channel with no flits to transmit. Future work would involve perfecting the HDL model and testing it with standard benchmarks.

4 Conclusion

This paper presented a dynamic technique to decrease packet starvation while using virtual channels is NoC routers. The technique involved splitting the operating time of the router into time slots and then assigning those to each of the virtual channels based on requirement. This allows a scalable performance enhancement system so that starvation of low priority packets happens only when the NoC is unavoidably congested. The paper discussed the technique in detail along with the prototype architecture and did a preliminary investigation of the performance merits and hardware overheads. Future work would involve modifying the prototype to full DTMVC specification and testing it with industry standard benchmarks.

5 References

[1] K. Goossens, J. Dielissen, and A. Radulescu, ‘AEthereal network on chip: concepts, architectures, and implementations’, *Des. Test Comput. IEEE*, vol. 22, no. 5, pp. 414 – 421, Oct. 2005.

[2] M. Millberg, E. Nilsson, R. Thid, S. Kumar, and A. Jantsch, ‘The Nostrum backbone-a communication protocol stack for Networks on Chip’, in *17th International Conference on VLSI Design, 2004. Proceedings*, 2004, pp. 693 – 696.

[3] A. Morgenshtein, A. Kolodny, and R. Ginosar, ‘Link Division Multiplexing (LDM) for Network-on-Chip Links’, in *2006 IEEE 24th Convention of Electrical and Electronics Engineers in Israel*, 2006, pp. 245 –249.

[4] W. J. Dally, ‘Virtual-channel flow control’, in *Proceedings of the 17th annual international symposium on Computer Architecture*, New York, NY, USA, 1990, pp. 60–68.

[5] T. Bjerregaard and J. Sparso, ‘A router architecture for connection-oriented service guarantees in the MANGO clockless network-on-chip’, in *Design, Automation and Test in Europe, 2005. Proceedings*, 2005, pp. 1226 – 1231 Vol. 2.

[6] A. Mello, L. Tedesco, N. Calazans, and F. Moraes, ‘Virtual channels in networks on chip: implementation and evaluation on hermes NoC’, in *Proceedings of the 18th annual symposium on Integrated circuits and system design*, New York, NY, USA, 2005, pp. 178–183.

[7] F. Ge, N. Wu, and Y. Wan, ‘A network monitor based dynamic routing scheme for Network on Chip’, in *Microelectronics Electronics, 2009. PrimeAsia 2009. Asia Pacific Conference on Postgraduate Research in*, 2009, pp. 133 –136.

[8] R. Manevich, I. Cidon, A. Kolodny, I. Walter, and S. Wimer, ‘A Cost Effective Centralized Adaptive Routing for Networks-on-Chip’, in *Digital System Design (DSD), 2011 14th Euromicro Conference on*, 2011, pp. 39 –46.

[9] V. Rantala, T. Lehtonen, P. Liljeberg, and J. Plosila, ‘Distributed Traffic Monitoring Methods for Adaptive Network-on-Chip’, in *NORCHIP, 2008.*, 2008, pp. 233 –236.

[10] Bharath Sudev, Leandro Soares Indrusiak, ‘PFT- A Low Overhead Predictability Enhancement Technique for Non-Preemptive NoCs’, in *21st IFIP/IEEE International Conference on Very Large Scale Integration (VLSI-SoC)*, Istanbul, Turkey, pp. 342 – 345.

[11] B. Sudev and L. S. Indrusiak, ‘Predictability Enhancement in Non-preemptive NoCs using Selective Packet Splitting’. 12th IEEE International Conference on Industrial Informatics (INDIN), Porto Alegre-Brazil, Jul-2014.

[12] B. Sudev and L. S. Indrusiak, ‘Low overhead predictability enhancement in non-preemptive network-on-chip routers using Priority Forwarded Packet Splitting’, in *2014 9th International Symposium on Reconfigurable and Communication-Centric Systems-on-Chip (ReCoSoC)*, 2014, pp. 1–8.

[13] F. Moraes, N. Calazans, A. Mello, L. Möller, and L. Ost, ‘HERMES: an infrastructure for low area overhead packet-switching networks on chip’, *Integr. VLSI J.*, vol. 38, no. 1, pp. 69–93, Oct. 2004.