modules a packet will traverse. As our evaluation shows, naively performing fair queuing on a single resource provides poor isolation for flows, violating the share-guarantee. Our attempt to extend WFQ by doing per-resource fair queuing (§4.2) turned out to violate strategy-proofness. Thus, DRFQ generalizes WFQ to multiple resources while providing isolation and strategy-proofness.

In the context of middleboxes, Egi *et al.* [14] proposed bottleneck fairness for software routers. We share their motivation for multi-resource fairness. However, we showed (§4.1) that their mechanism can not only provide poor isolation, but it can lead to heavy oscillations that severely degrade system performance. Dreger *et al.* [12] suggest measuring resource consumption of modules in NIDS and shutting off modules that overconsume resources. This approach is infeasible as some modules must run at all times, *e.g.,* a VPN module. Moreover, shutting down modules does not provide isolation between flows. With our approach, the flows that overconsume resources will fill buffers, eventually leading to modules not processing them, but each flow is sure to at least get its share guarantee of service.

In the context of active networks, Alexander *et al.* [7] propose a scheduling architecture called RCANE. This approach is akin to Per-Resource Fairness and therefore violates strategy-proofness.

Multi-resource fairness has also been investigated in the context of micro-economic theory. Ghodsi *et al.* [16] provide an overview and compare with the method preferred by economists, Competitive Equilibrium from Equal Incomes (CEEI). They show that CEEI is not strategy-proof and has several other undesirable properties. Dolev *et al.* [11] proposed an alternative to DRF. It too fails to be strategy-proof and is also computationally expensive to compute.

Our focus in this paper has been on achieving DRF allocations in the time domain. Others have analyzed how DRF allocations can be computed [19] and extended [21, 25] in the space domain.

## 10. CONCLUSION

Middleboxes apply complex processing functions to an increasing volume of traffic. Their performance characteristics are different from traditional routers; different processing functions have different demands across multiple resources, including CPU, memory bandwidth, and link bandwidth. Traditional single resource fair queuing schedulers therefore provide poor isolation guarantees between flows. Worse, in systems with multiple resources, flows can shift their demand to manipulate schedulers to get better service, thereby wasting resources. We have analyzed two schemes that are natural in the middlebox setting—bottleneck fairness and per-resource fairness—and shown that they have undesirable properties. In light of this, we have designed a new algorithm, DRFQ, for multi-resource fair queueing. We show through a Click implementation and extensive simulations that, unlike other approaches, our solution does not suffer from oscillations, provides flow isolation, and is strategy-proof. For future research directions, we believe DRFQ is applicable in many other multi-resource fair queueing contexts, such as VM scheduling in hypervisors.

## 11. ACKNOWLEDGEMENTS

We thank Adam Oliner, Ganesh Ananthanarayanan, and Patrick Wendell for useful feedback on earlier drafts of this paper. This research is supported in part by NSF FIA Award #CNS-1038695, NSF CISE Expeditions award CCF-1139158, a Google PhD Fellowship, gifts from Amazon Web Services, Google, SAP, Blue Goji, Cisco, Cloudera, Ericsson, General Electric, Hewlett Packard, Huawei, Intel, MarkLogic, Microsoft, NetApp, Oracle, Quanta, Splunk, VMware and by DARPA (contract #FA8650-11-C-7136).

## 12. REFERENCES


[1] Crossbeam network consolidation. `http://www.crossbeam.com/why-crossbeam/consolidation/`, June 2012.

[2] F5 Networks products. `http://www.f5.com/products/big-ip/`, June 2012.

[3] Intel perf. counter mon. `http://software.intel.com/en-us/articles/intel-performance-counter-monitor/`, June 2012.

[4] M57 network traffic traces. `https://domex.nps.edu/corp/scenarios/2009-m57/net/`, Feb. 2012.

[5] Palo alto networks. `http://www.paloaltonetworks.com/`, June 2012.

[6] Vyatta Software Middlebox. `http://www.vyatta.com`, June 2012.

[7] D. S. Alexander, P. B. Menage, A. D. Keromytis, W. A. Arbaugh, K. G. Anagnostakis, and J. M. Smith. The Price of Safety in An Active Network. *JCN*, 3(1):4–18, March 2001.

[8] K. Argyraki, K. Fall, G. Iannaccone, A. Knies, M. Manesh, and S. Ratnasamy. Understanding the packet forwarding capability of general-purpose processors. Technical Report IRB-TR-08-44, Intel Research Berkeley, May 2008.

[9] J. Bennett and H. Zhang. $WF^2Q$: Worst-case fair weighted fair queueing. In *INFOCOM*, 1996.

[10] A. Demers, S. Keshav, and S. Shenker. Analysis and simulation of a fair queueing algorithm. In *SIGCOMM*, pages 1–12, 1989.

[11] D. Dolev, D. G. Feitelson, J. Y. Halpern, R. Kupferman, and N. Linial. No justified complaints: on fair sharing of multiple resources. In *ITCS*, pages 68–75, 2012.

[12] H. Dreger, A. Feldmann, V. Paxson, and R. Sommer. Operational experiences with high-volume network intrusion detection. In *ACM Conference on Computer and Communications Security*, pages 2–11, 2004.

[13] H. Dreger, A. Feldmann, V. Paxson, and R. Sommer. Predicting the resource consumption of network intrusion detection systems. In *RAID*, 2008.

[14] N. Egi, A. Greenhalgh, M. Handley, G. Iannaccone, M. Manesh, L. Mathy, and S. Ratnasamy. Improved forwarding architecture and resource management for multi-core software routers. In *NPC*, pages 117–124, 2009.

[15] A. Ghodsi, V. Sekar, M. Zaharia, and I. Stoica. Multi-resource fair queueing for packet processing. Technical Report UCB/EECS-2012-166, EECS Department, University of California, Berkeley, June 2012.

[16] A. Ghodsi, M. Zaharia, B. Hindman, A. Konwinski, I. Stoica, and S. Shenker. Dominant resource fairness: Fair allocation of multiple resource types. In *NSDI*, 2011.

[17] S. J. Golestani. A self-clocked fair queueing scheme for broadband applications. In *INFOCOM*, pages 636–646, 1994.

[18] P. Goyal, H. Vin, and H. Cheng. Start-time fair queuing: A scheduling algorithm for integrated services packet switching networks. *ACM Transactions on Networking*, 5(5):690–704, Oct. 1997.

[19] A. Gutman and N. Nisan. Fair Allocation Without Trade. In *AAMAS*, June 2012.

[20] M. Honda, Y. Nishida, C. Raiciu, A. Greenhalgh, M. Handley, and H. Tokuda. Is it still possible to extend TCP? In *Proc. IMC*, 2011.

[21] C. Joe-Wong, S. Sen, T. Lan, and M. Chiang. Multi-resource allocation: Fairness-efficiency tradeoffs in a unifying framework. In *INFOCOM*, 2012.

[22] E. Kohler, R. Morris, B. Chen, J. Jannotti, and M. F. Kaashoek. The Click modular router. *ACM Trans. Comput. Syst.*, 18, August 2000.

[23] M. Kounavis, X. Kang, K. Grewal, M. Eszenyi, S. Gueron, and D. Durham. Encrypting the Internet. In *Proc. SIGCOMM*, 2010.

[24] A. Parekh and R. Gallager. A generalized processor sharing approach to flow control - the single node case. *ACM Transactions on Networking*, 1(3):344–357, June 1993.

[25] D. C. Parkes, A. D. Procaccia, and N. Shah. Beyond Dominant Resource Fairness: Extensions, Limitations, and Indivisibilities. In *ACM Conference on Electronic Commerce*, 2012.

[26] M. Piatek, T. Isdal, T. Anderson, A. Krishnamurthy, and A. Venkataramani. Do incentives build robustness in bittorrent. In *NSDI'07*, 2007.

[27] V. Sekar, N. Egi, S. Ratnasamy, M. Reiter, and G. Shi. Design and implementation of a consolidated middlebox architecture. In *NSDI*, 2012.

[28] V. Sekar, S. Ratnasamy, M. Reiter, N. Egi, and G. Shi. The Middlebox Manifesto: Enabling Innovation in Middlebox Deployments. In *HotNets 2011*, Oct. 2011.

[29] M. Shreedhar and G. Varghese. Efficient fair queuing using deficit round robin. *ACM Transactions on Networking*, 4(3):375–385, 1996.

[30] R. Smith, N. Goyal, J. Ormont, K. Sankaralingam, and C. Estan. Signature Matching in Network Processing Using SIMD/GPU Architectures. In *Int. Symp. on Performance Analysis of Systems and Software*, 2009.

[31] C. A. Waldspurger. *Lottery and Stride Scheduling: Flexible Proportional Share Resource Management*. PhD thesis, MIT, Laboratory of Computer Science, Sept. 1995. MIT/LCS/TR-667.

[32] Z. Wang, Z. Qian, Q. Xu, Z. M. Mao, and M. Zhang. An Untold Story of Middleboxes in Cellular Networks. In *SIGCOMM*, 2011.

[33] L. Zhang. Virtual clock: a new traffic control algorithm for packet switching networks. *SIGCOMM CCR*, 20:19–29, August 1990.