MAROQ: Um Modelo de Alocação de Recursos Orientado a Qualidade de Experiência

André L. DAmato, Mário Dantas

[1]Departamento de Informática e Estatística – Universidade Federal de Santa Catarina (UFSC)
Santa Catarina – SC – Brasil

{altdamato@gmail.com, mario.dantas@ufsc.br}

Abstract. *The growing trend of cloud computing in network has generated challenges to allocate resources. The quality of experience arises as a important paradigm to solving many challenges. The quality experience takes into account a series of context parameters. Therefore, it is proposed in this paper the resources allocation model MAROQ, which is a quality of experience driven-model that uses context information to resource allocation in clounds, and computational grids. Experimental results show that using context information improves the performance of the task submission.*

Resumo. *A quantidade de recursos fornecidos pelas nuvens computacionais na Internet gerou desafios complexos para resolver problemas relacionados a alocação desses recursos. A qualidade de experiência surge como um paradigma diferenciado como uma fator potencialmente importante na solução desse desafios. A qualidade de experiência leva em consideração parâmetros de contexto. Sendo assim, é proposto neste trabalho o modelo de alocação de recursos MAROQ, que é um modelo orientado a qualidade de experiência, que utiliza informações de contexto para alocação de recursos em nuvens e grids computacionais. Resultados experimentais mostram que utilizar informações de contexto melhora o desempenho na submissão de tarefas.*

1. Introdução

Com a crescente quantidade de nuvens computacionais disponibilizadas na Internet, a utilização de *big data* torna-se um desafio complexo. Esse desafio esta relacionado com a maneira a qual os recursos serão acessados nas nuvens. A modelagem desse tipo de problema muitas vezes esta próxima do uso da teoria do caos [Alkhatib and Krunz 2000] para algumas soluções. Este cenário dificulta a utilização correta dos recursos pelos usuários. A qualidade de um serviço é percebida pelos usuários por fatores cognitivos. Esta percepção subjetiva de qualidade é conhecida como qualidade de experiência (ou em inglês *quality of experience*) (QoE) [Shaikh et al. 2010]. Sendo assim, a maneira com que os recursos são acessados é uma questão determinante para uma boa qualidade de experiência sobre os serviços disponibilizados pelas nuvens computacionais.

Portanto, a qualidade de experiência surge como um paradigma diferenciado e particularmente importante para a alocação de recursos, uma vez que uma boa qualidade de alocação de recursos deve levar em consideração uma serie de parâmetros de contexto [Názario and Dantas 2012] (como por exemplo, tipo de processador, sistema operacional, quantidade de memória). Essas informações de contexto podem ser fornecidas pelo usuário, pois o mesmo conhece o contexto apropriado para execução de sua aplicação.

Sendo assim, este trabalho apresenta o *MAROQ* (Modelo de Alocação de Recursos Orientado a Qualidade de experiência). O modelo proposto visa a alocação de recursos orientada a QoE, considerando o contexto de aplicação. O modelo proposto tem como foco execução de aplicações em nuvens computacionais formadas por *clusters* e *grids*.

O restante do texto é organizado da seguinte maneira: a Seção 2 aborda os principais modelos de QoE propostos nos últimos anos; as Seções 3, e 4 abordam os conceitos básicos para o modelo MAROQ proposto na Seção 5; a seção 6 aborda resultados prévios, e por fim na Seção 7 são apresentadas as considerações finais, e são sugeridas algumas diretivas para trabalhos futuros.

2. Trabalhos Relacionados

Como mencionado anteriormente na seção introdutória, os escalonadores estáticos alocam uma fatia fixa de tempo para um determinado *job*. Diversos modelos como os propostos em [Chase et al. 2003, Sotomayor et al. 2008] utilizam escalonadores estáticos baseados em reserva de recursos. Chase propõe em seu trabalho uma arquitetura flexível para gerenciamento de *grids*. A arquitetura de Chase é considerada flexível por permitir que novos nós sejam configurados dinamicamente de acordo com a disponibilidade dos recursos fornecidos por uma *grid*. No modelo proposto por Chase os recursos são atendidos conforme a demanda. Para garantir que os recursos sejam liberados dinamicamente são realizados monitoramentos nas cargas de trabalho impostas aos conjuntos de nós agrupados logicamente em clusters virtuais (Vclusters) [Chase et al. 2003].

O trabalho de Micillo [Micillo et al. 2009] propõe uma implementação de mecanismos para otimização na alocação de recursos também no contexto de *grids* computacionais. A arquitetura proposta por Micillo possui um gerenciador centralizado, ao qual, é responsável por gerenciar os recursos intergrid. Embora a arquitetura proposta por Micillo suporte migração de jobs, esta funcionalidade depende de compilação utilizando bibliotecas específicas. Hermenier [Hermenier et al. 2010] avança um pouco mais e propõe alocação dinâmica de recursos utilizando uma abstração baseada em transição de estados (executando, esperando, suspenso) para os *jobs*. Assim como Chase e Sotomayor, Hermenier também utiliza máquinas virtuais para compor seu modelo. Neste trabalho foi definida uma abstração de *jobs*, sendo cada *job* entendido como a composição de diversas máquinas virtuais. Estes *jobs* são chamados de *jobs* virtuais. Os *jobs* virtuais possuem 4 estados de execução sendo eles: executando, dormindo, esperando, e terminado. Embora apresentem estratégias para implementação de políticas de alocação de recursos, os trabalhos mencionados anteriormente não proveem mecanismos de contexto para tomada de decisões no momento de alocar recursos. Visando justiça para alocação de recursos, [Ghodsi et al. 2011] propôs em seu trabalho o modelo DFS (*Dominant Resource Fairness*, ou Recursos Dominantes e Justiça) de escalonamento de tarefas para ambientes *MapReduce*. O foco do trabalho de Ghodsi é favorecer o compartilhamento de recursos considerando quais são os recursos dominantes requisitados por uma aplicação. Nesse contexto, recursos dominantes são os recursos que uma determinada aplicação mais depende para executar, por exemplo, aplicações orientadas a processamento demandam mais de CPUs, e memória são prioridade para aplicações orientadas a funções de entrada e saída. No entanto, os recursos não são ajustados com foco nas aplicações, e sim com foco no compartilhamento com o objetivo de obter a máxima utilização dos recursos.

Joseph [Joseph et al. 2012] e Zhou [Zhou et al. 2013] apresentam uma formalização matemática para relacionar alocação de recursos e justiça com qualidade de experiência. O cenário proposto por Zhou remete a redes sem fio multimídia cegas, ou seja, redes sem fio ao qual as estações bases não tem conhecimento sobre a modelagem de qualidade de experiência para o conjunto de aplicações que utilizam o sistema. No entanto ao contrário de Joseph, Zhou não estabelece em sua formalização a relação entre a degradação da qualidade de experiência e a variabilidade da alocação dos recursos com o tempo. Segundo Zhou, a qualidade de experiência varia de acordo com a quantidade de usuários utilizando o sistema e o fator α, que define o grau de justiça para cada usuário quando o mesmo deseja alocar um determinado recurso.

3. Qualidade de experiência e contexto

Existem diferenças de percepção de desempenho sobre um determinado serviço de internet considerando o ponto de vista do usuário e o ponto de vista técnico abordado pelos provedores de serviço [Shaikh et al. 2010]. Isto acontece pelo fato dos provedores utilizarem métrica referentes a qualidade de serviço, ou em inglês *Quality of Service* (QoS), sendo estas métricas estabelecidas a partir de dados técnicos como por exemplo rendimento da rede, vazão, perda de pacotes, atraso etc. Estes parâmetros são geralmente medidos por meio de computadores inseridos na rede que monitoram o desempenho da mesma. No entanto, os usuários percebem o desempenho da rede seguindo métricas mais subjetivas e não técnicas. Esta percepção subjetiva é conhecida como qualidade de experiência, ou em inglês *Quality of Experience* (QoE) [Shaikh et al. 2010].

Para avaliar a percepção do usuário sobre um determinado serviço disponibilizado pela rede, são utilizados questionários aplicados em ambientes controlados. Este tipo de teste é conhecido pela comunidade como pontuação média de opinião, ou em inglês *Mean Opinion Score* (MOS). Esta técnica é geralmente aplicada para avaliar sistemas multimídias. No entanto, devido ao grande número e a diversidade de aplicações testes de opinião não são mais a melhor alternativa. Além disso, os testes de opinião são muito criticados por autores de várias áreas [De Moor et al. 2010]. Estas críticas estão relacionadas a escala de pontuação adotada nos testes, sendo estas consideradas por alguns autores como por exemplo [Koning et al. 2007] como escalas imprecisas e não representativas considerando diferenças culturais de interpretação. Segundo [Sullivan et al. 2008], esta escala determina valores absolutos obtidos em ambientes controlados, ao qual não representam com precisão ambientes reais pelo fato de não considerar a influência de variáveis de contexto.

Contexto é um conjunto de informações que descrevem a situação de uma determinada entidade, que por sua vez pode ser definido como qualquer componente relevante para interação entre uma aplicação e seus respectivos usuário [Dey 2000]. Um sistema é sensível ao contexto (ou em inglês *context-aware*), se este utiliza informações de contexto para prover informações, ou serviços relevantes para o usuário, sendo que a relevância depende diretamente da tarefa exercida pelo usuário dentro do sistema [Dey 2000]. Em outras palavras, as informações de contexto são utilizadas para prover informações, e serviços ajustados as necessidades do usuário. Desta maneira é possível aumentar a satisfação do usuário na utilização do sistema, melhorando consequentemente a qualidade de experiência. Uma boa modelagem de contexto permite também diminuir a complexidade de aplicações computacionais sensíveis ao contexto (context-aware appli-

cations [Schilit et al. 1994]). O trabalho de Bettini [Bettini et al. 2010] é direcionado a computação pervasiva e aborda diversas modelagens de contexto.

4. Alocação de recursos em nuvens

Em ambientes distribuídos os recursos computacionais são determinados por uma cadeia de recursos aleatórios (*network queue theory*). Segundo [Stolyar 2005] os recursos em uma rede computacional necessitam de uma modelagem randômica para utilização dos recursos. Ou seja, os mecanismos de controle de recursos formam uma cadeia de Markov, ao qual as transições de estados não dependem do estado anterior. Isto se deve ao fato da utilização de recursos pelos usuários acontecer de maneira aleatória dentro de uma rede computacional. O escalonamento de recursos é realizado por meio de escalonadores que podem utilizar diferentes políticas de escalonamento. As estratégias mais comuns definidas para os escalonadores de recursos são as seguintes: O FCFS (*First Come First Served*), onde as primeiras tarefas que chegam são as primeiras servidas; *Backfiling* ao qual tenta resolver problemas de fragmentação; Preempção, que melhora os resultados do *backfiling* utilizando informações da aplicação. O método FCFS resulta na fragmentação de recursos, ou seja, diversos recursos se tornam obsoletos no momento da execução das tarefas. O método*backfiling* não gerencia os recursos quando as aplicações mudam seus requisitos dinamicamente. Os mecanismos preemptivos garantem a execução das tarefas de acordo com suas características (por exemplo: prioridade, requisitos, tempo de execução). Para [Joseph et al. 2012], a qualidade de experiência é definida como uma métrica diretamente associada a qualidade de alocação, a variabilidade e a justiça no momento da alocação de recursos. Portanto é possível melhorar a qualidade de experiência de um usuário ajustando estes parâmetros definidos para alocação de recursos.

5. O Modelo MAROQ

O modelo proposto tem como objetivo otimizar a utilização de recursos em nuvens computacionais, visando a satisfação do usuário quanto a utilização dos recursos fornecidos pelo ambiente. O modelo MAROQ visa obter a melhor qualidade de experiência resultante a partir do gerenciamento dos parâmetros: significância, variabilidade e justiça na alocação de recursos. O plano formado pelos parâmetros citados determinam a qualidade de experiência [Joseph et al. 2012]. Neste sentido, é proposto que o modelo MAROQ seja orientado a contexto para realizar a melhor alocação de recursos possível ajustando assim o parâmetro de significância para que os recursos não precisem ser realocados durante a execução de uma aplicação. Além disso, o contexto permite a otimização do espaço de busca por recursos melhorando assim o desempenho para realizar a alocação dos mesmos.

Como mencionado anteriormente, a QoE em um sistema computacional segundo [Joseph et al. 2012] é a composição dos parâmetros alocação ajustada *mean*, variabilidade e justiça. Seguindo esta definição é proposto no modelo MAROQ a utilização de contexto para lidar com a alocação ajustada e com a variabilidade na alocação de recursos. Em outras palavras, a utilização de contexto permite alocar recursos de maneira que os mesmos não precisem ser realocados posteriormente. A justiça na alocação de recursos é realizada a partir de preempção de tarefas. O mecanismo de preempção garante que uma tarefa de menor prioridade, libere os seus recursos no momento que uma tarefa de maior prioridade chega.

O *middleware* [Scheidt et al. 2012] implementado no modelo é composto por um mecanismo que aloca recursos de acordo com a prioridade de cada tarefa, lidando assim com o parâmetro justiça. A Figura 1 ilustra o modelo MAROQ. O plano de controle resultante visa atuar como gerenciador de clusters em nuvens computacionais, focando sempre em obter a melhor QoE possível.

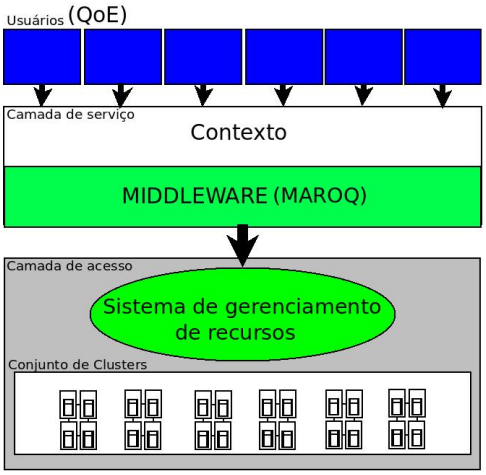

Figura 1. Modelo proposto

Na Figura 1 a parte superior ilustra os usuários dos recursos disponíveis pelos clusters, sendo estes responsáveis pela requisição dos recursos inserindo informações de contexto. As informações de contexto são referentes ao tipo de processador, sistema operacional, quantidade de memória necessária. A partir destas informações o MAROQ imediatamente realiza a síntese de um arquivo de submissão, que é utilizado como entrada para o sistema de gerenciamento de recursos. O modelo utiliza o balanceador de carga baseado no próprio gerenciador de recursos.

Uma outra informação de contexto importante considerada no modelo é inerente a quantidade de comunicação entre os processos, pois esta informação revela a necessidade de desempenho dos componentes de interconexão intra e inter-cluster. Ou seja, quanto maior a quantidade de comunicação entre os processos da aplicação, maior deverá ser a velocidade do componente(s) de interconexão do(s) cluster(s) alocado(s). Pois a latência de comunicação é menor entre processos dentro de um mesmo cluster, do que em processos alocados em clusters geograficamente distribuídos. Por outro lado, se uma aplicação é composta por processos que não se comunicam esta pode ser alocada em clusters distintos. A Figura 2 mostra a arquitetura do sistema MAROQ. A parte superior da figura representa o usuário, que realiza a requisição de execução de uma aplicação passando o contexto desejado para esta execução. Uma vez determinada o contexto e realizada a requisição esta é enviada via *web service* [Kalin 2013] utilizando o protocolo SOAP [Ryman 2001]. A requisição realizada pelo usuário é recebida pelo meta escalonador do MAROQ. O meta escalonado realiza assim uma negociação com os lideres de cada *cluster* a fim de identificar o que possui o melhor conjunto de recursos para atender ao contexto especificado. Após o *cluster* que atenda aos requisitos ser identificado, o líder deste cluster responsável por executar o controle do gerenciamento de recursos recebe via *web service* os dados para execução da tarefa a ser submetida. Os nós de cada *cluster* são responsáveis apenas por executar as aplicações.

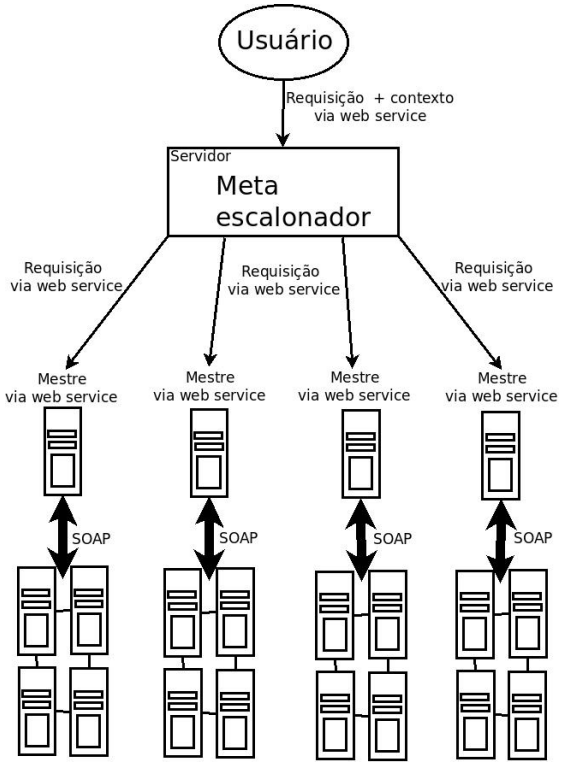

Figura 2. Arquitetura do MAROQ

6. Resultados

O gráfico da Figura 3, mostra que as submissões que utilizam informações de contexto possuem maior desempenho em relação a tempo, quando comparadas as submissões que não utilizam contexto.

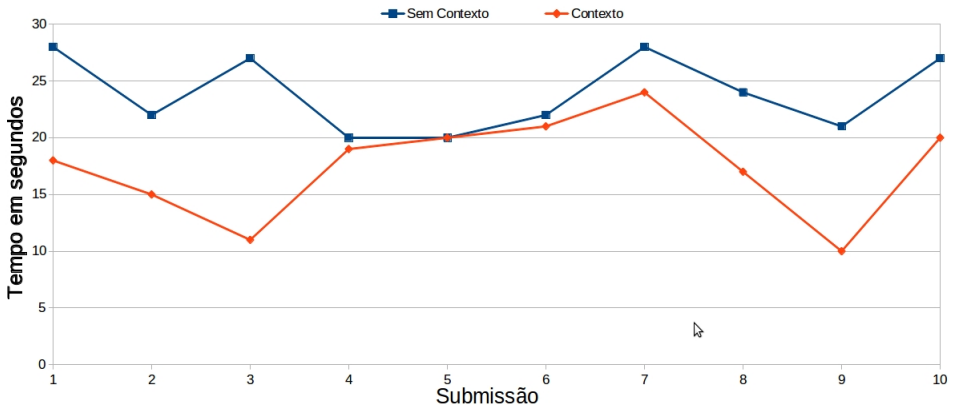

Figura 3. Resultados

O experimento foi realizado em 10 máquinas virtuais com 1 núcleo cada uma utilizando a ferramenta VmWare. O sistema de gerenciamento utilizado neste experimento foi o Condor [Frey et al. 2001]. A aplicação submetida é composta por um *sleeper*, que recebe o tempo que a aplicação ficará dormindo como parâmetro na invocação do programa. Para todas as submissões o parâmetro utilizado foi o mesmo. As informações

de contexto consideradas foram as seguintes: número de cpus, quantidade de memória, tamanho de disco rígido e sistema operacional. A aplicação foi escolhida pelo fato da utilização da função *sleep* permitir isolar os tempos gastos com a aplicação e com a alocação de recurso, sendo que é conhecido o tempo despendido com a função *sleep*. No caso da aplicação submetida ao experimento o parâmetro utilizado foi 5 segundos. Ou seja, dos tempos mencionados no gráfico da Figura 3 é possível dizer que aproximadamente 5 segundos foram gastos com a função *sleep*, sendo o tempo restante utilizado para as demais rotinas da aplicação e do sistema de alocação de recursos. Os resultados obtidos com o experimento mostram que inserir informações de contexto diminui o tempo para realizar a submissão de tarefas. Isto ocorre pela diminuição do espaço de busca por recursos.

7. Conclusão e Trabalhos Futuros

Este trabalho apresenta o modelo MAROQ, sendo este um modelo de alocação de recursos para nuvens computacionais orientado a QoE e contexto. Informações de contexto permitem que os recursos sejam ajustados as necessidades da aplicação. Desta maneira é possível melhorar o desempenho do sistema e consequentemente melhorar a QoE do usuário. Os experimentos realizados revelaram que o modelo permite melhorar o desempenho na alocação de recursos. Isso ocorre pelo ajuste e diminuição do espaço de busca de recursos. Como as informações de contexto ajustam os recursos de acordo com a aplicação, acredita-se que o MAROQ possua potencial para melhorar também o desempenho quanto ao tempo de execução. Sendo assim, é pretendido como trabalho futuro realizar experimentos focando no tempo de execução utilizando informações de contexto.

Referências

Alkhatib, A. and Krunz, M. (2000). Application of chaos theory to the modeling of compressed video. In *Communications, 2000. ICC 2000. 2000 IEEE International Conference on*, volume 2, pages 836–840 vol.2.

Bettini, C., Brdiczka, O., Henricksen, K., Indulska, J., Nicklas, D., Ranganathan, A., and Riboni, D. (2010). A survey of context modelling and reasoning techniques. *Pervasive Mob. Comput.*, 6(2):161–180.

Chase, J. S., Irwin, D. E., Grit, L. E., Moore, J. D., and Sprenkle, S. E. (2003). Dynamic virtual clusters in a grid site manager. In *Proceedings of the 12th IEEE International Symposium on High Performance Distributed Computing*, HPDC '03, pages 90–, Washington, DC, USA. IEEE Computer Society.

De Moor, K., Ketyko, I., Joseph, W., Deryckere, T., De Marez, L., Martens, L., and Verleye, G. (2010). Proposed framework for evaluating quality of experience in a mobile, testbed-oriented living lab setting. *Mobile Networks and Applications*, 15(3):378–391.

Dey, A. K. (2000). *Providing Architectural Support for Building Context-aware Applications*. PhD thesis, Atlanta, GA, USA. AAI9994400.

Frey, J., Tannenbaum, T., Livny, M., Foster, I., and Tuecke, S. (2001). Condor-g: a computation management agent for multi-institutional grids. In *High Performance Distributed Computing, 2001. Proceedings. 10th IEEE International Symposium on*, pages 55–63.

Ghodsi, A., Zaharia, M., Hindman, B., Konwinski, A., Shenker, S., and Stoica, I. (2011). Dominant resource fairness: Fair allocation of multiple resource types. In *NSDI*, volume 11, pages 24–24.

Hermenier, F., Lèbre, A., and Menaud, J.-M. (2010). Cluster-wide context switch of virtualized jobs. In *Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing*, HPDC '10, pages 658–666, New York, NY, USA. ACM.

Joseph, V., de Veciana, G., and Arapostathis, A. (2012). Resource allocation: Realizing mean-variability-fairness tradeoffs. In *Allerton Conference*, pages 831–838. IEEE.

Kalin, M. (2013). *Java Web Services: Up and Running*. O'Reilly Media, Inc.

Koning, T. C. M., Veldhoven, P., Knoche, H., and Kooij, R. E. (2007). Of MOS and men: bridging the gap between objective and subjective quality measurements in mobile tv. In *SPIE*.

Micillo, R., Venticinque, S., Aversa, R., and Di Martino, B. (2009). A grid service for resource-to-agent allocation. In *Internet and Web Applications and Services, 2009. ICIW '09. Fourth International Conference on*, pages 443–448.

Názario, D. C. and Dantas, M. A. R. (2012). Taxonomia das publicações sobre qualidade de contexto. In *International Journal of Sustainable business*, number 20, pages 1–28.

Ryman, A. (2001). Simple object access protocol SOAP and web services. In *Proceedings of the 23rd International Conference on Software Engineering*, ICSE '01, pages 689–, Washington, DC, USA. IEEE Computer Society.

Scheidt, R. F., Schmidt, K., Pessoa, G. M., Viera, M. A., and Dantas, M. A. R. (2012). A software product line approach to enhance a meta-scheduler middleware. In *Journal of Physics: Conference Series*, volume 341, pages 1–7.

Schilit, B., Adams, N., and Want, R. (1994). Context-aware computing applications. In *Proceedings of the 1994 First Workshop on Mobile Computing Systems and Applications*, WMCSA '94, pages 85–90, Washington, DC, USA. IEEE Computer Society.

Shaikh, J., Fiedler, M., and Collange, D. (2010). Quality of experience from user and network perspectives. *annals of telecommunications*, 65(1-2):47–57.

Sotomayor, B., Keahey, K., and Foster, I. (2008). Combining batch execution and leasing using virtual machines. In *Proceedings of the 17th International Symposium on High Performance Distributed Computing*, HPDC '08, pages 87–96, New York, NY, USA. ACM.

Stolyar, A. L. (2005). Maximizing queueing network utility subject to stability: Greedy primal-dual algorithm. *Queueing Syst. Theory Appl.*, 50(4):401–457.

Sullivan, M., Pratt, J., and Kortum, P. (2008). Practical issues in subjective video quality evaluation: Human factors vs. psychophysical image quality evaluation. In *Proceedings of the 1st International Conference on Designing Interactive User Experiences for TV and Video*, UXTV '08, pages 1–4, New York, NY, USA. ACM.

Zhou, L., Chen, M., Qian, Y., and Chen, H.-H. (2013). Fairness resource allocation in blind wireless multimedia communications. *Multimedia, IEEE Transactions on*, 15(4):946–956.