OULU 2017
C 613

ACTA UNIVERSITATIS OULUENSIS

Maria Kangas

STABILITY ANALYSIS OF NEW PARADIGMS IN WIRELESS NETWORKS

UNIVERSITY OF OULU GRADUATE SCHOOL;
UNIVERSITY OF OULU,
FACULTY OF INFORMATION TECHNOLOGY AND ELECTRICAL ENGINEERING;
CENTRE FOR WIRELESS COMMUNICATIONS;
INFOTECH OULU

ACTA UNIVERSITATIS OULUENSIS
C Technica 613

MARIA KANGAS

STABILITY ANALYSIS OF NEW PARADIGMS IN WIRELESS NETWORKS

Academic dissertation to be presented with the assent of the Doctoral Training Committee of Technology and Natural Sciences of the University of Oulu for public defence in Kuusamonsali (YB210), Linnanmaa, on 12 June 2017, at 12 noon

UNIVERSITY OF OULU, OULU 2017

Acta Univ. Oul. C 613, 2017

Supervised by
Professor Savo Glisic

Reviewed by
Professor Luis M. Correia
Professor Francisco Javier Conzales Castano

Opponent
Professor Evgeny Kucheryavy

ISBN 978-952-62-1545-7 (Paperback)
ISBN 978-952-62-1546-4 (PDF)

ISSN 0355-3213 (Printed)
ISSN 1796-2226 (Online)

Cover Design
Raimo Ahonen

JUVENES PRINT
TAMPERE 2017

Kangas, Maria, Stability analysis of new paradigms in wireless networks.
University of Oulu Graduate School; University of Oulu, Faculty of Information Technology and Electrical Engineering; Centre for Wireless Communications; Infotech Oulu
Acta Univ. Oul. C 613, 2017
University of Oulu, P.O. Box 8000, FI-90014 University of Oulu, Finland

Abstract

Fading in wireless channels, the limited battery energy available in wireless handsets, the changing user demands and the increasing demand for high data rate and low delay pose serious design challenges in the future generations of mobile communication systems. It is necessary to develop efficient transmission policies that adapt to changes in network conditions and achieve the target delay and rate with minimum power consumption.

In this thesis, a number of new paradigms in wireless networks are presented. Dynamic programming tools are used to provide dynamic network stabilizing resource allocation solutions for virtualized data centers with clouds, cooperative networks and heterogeneous networks. Exact dynamic programming is used to develop optimal resource allocation and topology control policies for these networks with queues and time varying channels. In addition, approximate dynamic programming is also considered to provide new sub-optimal solutions.

Unified system models and unified control problems are also provided for both secondary service provider and primary service provider cognitive networks and for conventional wireless networks. The results show that by adapting to the changes in queue lengths and channel states, the dynamic policy mitigates the effects of primary service provider and secondary service provider cognitive networks on each other.

We investigate the network stability and provide new unified stability regions for primary service provider and secondary service provider cognitive networks as well as for conventional wireless networks. The K-step Lyapunov drift is used to analyse the performance and stability of the proposed dynamic control policies, and new unified stability analysis and queuing bound are provided for both primary service provider and secondary service provider cognitive networks and for conventional wireless networks. By adapting to the changes in network conditions, the dynamic control policies are shown to stabilize the network and to minimize the bound for the average queue length. In addition, we prove that the previously proposed frame based does not minimize the bound for the average delay, when there are shared resources between the terminals with queues.

Keywords: access point, ad hoc network, cooperative communication, dynamic programming, lyapunov drift, network stability, topology control, value iteration algorithm

Kangas, Maria, Uusien paradigmojen stabiiliusanalyysi langattomissa verkoissa.
Oulun yliopiston tutkijakoulu; Oulun yliopisto, Tieto- ja sähkötekniikan tiedekunta; Centre for Wireless Communications; Infotech Oulu
Acta Univ. Oul. C 613, 2017
Oulun yliopisto, PL 8000, 90014 Oulun yliopisto

Tiivistelmä

Langattomien kanavien häipyminen, langattomien laitteiden akkujen rajallinen koko, käyttäjien käyttötarpeiden muutokset sekä lisääntyvän tiedonsiirron ja lyhyemmän viiveen vaatimukset luovat suuria haasteita tulevaisuuden langattomien verkkojen suunnitteluun. On välttämätöntä kehittää tehokkaita resurssien allokointialgoritmeja, jotka sopeutuvat verkkojen muutoksiin ja saavuttavat sekä tavoiteviiveen että tavoitedatanopeuden mahdollisimman pienellä tehon kulutuksella.

Tässä väitöskirjassa esitetään uusia paradigmoja langattomille tietoliikenneverkoille. Dynaamisen ohjelmoinnin välineitä käytetään luomaan dynaamisia verkon stabiloivia resurssien allokointiratkaisuja virtuaalisille pilvipalveludatakeskuksille, käyttäjien yhteistyöverkoille ja heterogeenisille verkoille. Tarkkoja dynaamisen ohjelmoinnin välineitä käytetään kehittämään optimaalisia resurssien allokointi ja topologian kontrollointialgoritmeja näille jonojen ja häipyvien kanavien verkoille. Tämän lisäksi, estimoituja dynaamisen ohjelmoinnin välineitä käytetään luomaan uusia alioptimaalisia ratkaisuja.

Yhtenäisiä systeemimalleja ja yhtenäisiä kontrollointiongelmia luodaan sekä toissijaisen ja ensisijaisen palvelun tuottajan kognitiivisille verkoille että tavallisille langattomille verkoille. Tulokset osoittavat että sopeutumalla jonojen pituuksien ja kanavien muutoksiin dynaaminen tekniikka vaimentaa ensisijaisen ja toissijaisen palvelun tuottajien kognitiivisten verkkojen vaikutusta toisiinsa.

Tutkimme myös verkon stabiiliutta ja luomme uusia stabiilisuusalueita sekä ensisijaisen ja toissijaisen palveluntuottajan kognitiivisille verkoille että tavallisille langattomille verkoille. K:n askeleen Lyapunovin driftiä käytetään analysoimaan dynaamisen kontrollointitekniikan suorituskykyä ja stabiiliutta. Lisäksi uusi yhtenäinen stabiiliusanalyysi ja jonon yläraja luodaan ensisijaisen ja toissijaisen palveluntuottajan kognitiivisille verkoille ja tavallisille langattomille verkoille. Dynaamisen algoritmin näytetään stabiloivan verkko ja minimoivan keskimääräisen jonon pituuden yläraja sopeutumalla verkon olosuhteiden muutoksiin. Tämän lisäksi todistamme että aiemmin esitetty frame-algoritmi ei minimoi keskimääräisen viiveen ylärajaa, kun käyttäjät jakavat keskenään resursseja.

Asiasanat: access point, ad hoc-verkko, arvoiteraatioalgoritmi, dynaaminen ohjelmointi, lyapunov drift, topologian kontrollointi, verkon stabiilius, yhteistyö kommunikaatio

Preface

The research work presented in this thesis was carried out at the Department of Communications Engineering, Centre of Wireless Communications (CWC), University of Oulu, Finland, during the years 2008-2016.

I thank all my - past and present - superiors and colleagues who have helped me in my work towards this thesis. I thank my supervisor Professor Savo Glisic for the patience and guidance over these years. I also express my gratitude for the Heads of the Department of Communication Engineering for providing me this opportunity. I would also thank Professor Markku Juntti for his support within Infotech Oulu.

I am grateful to my reviewers Professor Luis Correia (Technical University of Lisbon, Portugal) and Professor Javier Castano (University of Vigo, Spain) for having the patience to review my thesis. I also want to thank Professor Kucheryavy Evgeny for agreeing to serve as opponent.

The main financial support for this work was provided by the Finnish Funding Agency for Technology and Innovation (Tekes), the Academy of Finland, Infotech Oulu Graduate School, Nokia and Nokia Siemens Networks. I have also received personal funding from HPY research foundation, TES research foundation and Riitta and Jorma J. Takanen foundation. These acknowledgements have been very encouraging and are gratefully recognized.

Very special thanks go to my partner Jari. Your support has been invaluable and advices appreciated.

Finally, many thanks for the whole CWC staff. Especially, I would like to thank my friends Beatriz Lorenzo, Emmi Kaivanto, Mariella Särestöniemi, Tuomo Hänninen, Matti Kangas and Animesh Yadav.

Oulu, November, 2016

Abbreviations

$\arg\max\{\cdot\}$	*Arguments of the maxima*
$\mathrm{Conv}\{\cdot\}$	*Convex combination*
$\mathbb{E}\{\cdot\}$	*Expected value operator*
$\mathbb{E}\{\cdot\|\cdot\}$	*Conditional expectation*
$\lim$	*Limit*
$\limsup$	*Limit superior*
$\log_2(\cdot)$	*Logarithm in base 2*
$\max\{\cdot\}$	*Maximum*
$\min\{\cdot\}$	*Minimum*
$\mathrm{p}\{\cdot\}$	*Probability that the given event occurs*
a_i	*Arrival rate at terminal i*
$a^{\max}$	*Maximum arrival rate*
AP_l	*lth access point*
$\vec{A}$	*Arrival rate vector*
b	*Number of bits in a packet*
b_{is}	*Binary variable*
$\vec{B}_i$	*Vector of binary variables*
c_i	*Long-term average service rate at terminal i*
c_{is}	*Long-term average service rate from terminal i to server s*
$C_i^{V_{ij}}$	*Channel capacity*
D_X	*Control input in state X*
$\mathscr{D}_X$	*Set of feasible control options in state X*
f	*Utilization level*
$f_{\min}$	*Minimun frequency*
$f_{\max}$	*Maximum frequency*
$\vec{G}$	*Vector of average long-term service rates*
g_i	*Total long-term average service rate at terminal i*
g_{i0}	*Long-term average service rate between terminal i and access point*
g_{ij}	*Long-term average service rate between terminal i and terminal j*
$\vec{G}^*$	*Vector of unified average long-term service rates*
h_{is}	*Channel gain between terminal i and server s*

h_{ij}	*Channel gain between terminal i and terminal j*
h_{i0}	*Channel gain between terminal i and access point*
h_{nl}	*Channel gain between terminal i and access point l*
$\vec{H}$	*Channel gain vector*
$\mathscr{H}$	*Channel state space*
$\mathscr{H}_i$	*Channel state space for terminal i*
$\vec{H}_i$	*Channel gain vector for terminal i*
$\vec{H}_n$	*Channel gain vector for terminal n*
$\vec{H}_{\mathrm{e}}$	*Equivalent channel gain vector*
$\vec{H}_i^{\mathrm{e}}$	*Equivalent channel gain vector*
I	*Channel availability indicator*
i	*Terminal index*
$\mathscr{I}$	*Set of terminals*
$\lvert\mathscr{I}\rvert$	*Number of terminals*
I_{r}	*Channel corruption indicator*
j	*Time index*
M	*Number of active access points*
m_{ij}	*Binary variable*
$\lvert\mathscr{M}\rvert$	*Number of cooperating pairs*
N	*Number of terminals*
n	*Frame index*
L	*Number of potential access points*
$\mathscr{L}$	*Set of potential access points*
$\vec{P}$	*Power vector*
P_i	*Power consumption at terminal i*
p_j	*Probability*
P_{is}	*Power required for transmission from terminal i to server s*
p_{id}	*The probability of secondary user to detect the idling channel*
p_{pd}	*The probability of secondary user to detect the preamble correctly*
p_{sd}	*The probability of secondary user to detect the presence of primary user*
$\hat{P}_s$	*Power consumption at server s*
$P^{\max}$	*Maximum power available at terminal i*
$\hat{P}^{\max}$	*Maximum power available at terminal s*
P_i^{tot}	*Total power consumption at terminal i*

p_H^P	*Channel availability probability for primary service provider cognitive network*
p_{return}^P	*The probability of primary user to return to the channel*
p_H^S	*Channel availability probability for secondary service provider cognitive network*
p_{return}^S	*The probability of secondary user to return to the channel*
p_0^P	*Channel non-availability probability for primary service provider cognitive network*
p_0^S	*Channel non-availability probability for secondary service provider cognitive network*
p_1^P	*The probability of primary user to be active*
p_1^S	*The probability of secondary user to use the channel*
q_i	*Queue length at terminal i*
$q^{\max}$	*Maximum queue length*
$\hat{q}_s^i$	*Queue length of terminal i at server s*
$\vec{Q}$	*Vector of queue lengths*
r_{nm}	*Service rate from terminal to an access point*
$\vec{r}$	*Service rate vector*
s	*Server index*
$\mathscr{S}$	*Set of servers*
$\lvert\mathscr{S}\rvert$	*Number of servers*
t_0	*Time index*
U_i	*Control action at terminal i*
U_n	*nth terminal*
U_{X_i}	*Control action at terminal i in state X_i*
$\vec{U}$	*Vector of control action at terminal i*
$\mathscr{U}$	*Set of control actions at terminals*
$\hat{\mathscr{U}}$	*Set of control actions at servers*
$\hat{U}$	*Vector of control action at server s*
$\mathscr{U}_{X_i}$	*Set of control actions at terminal i in state X_i*
$\mathscr{U}_X$	*Set of control actions in state X*
$\lvert\mathscr{U}_{X_i}\rvert$	*Number of control actions in a state at terminal i*
$\hat{U}_{\hat{X}_s}$	*Control action at server s in state $\hat{X}_s$*
$\hat{U}_s$	*Control action at server s*
$\lvert\hat{\mathscr{U}}_{\hat{X}_s}\rvert$	*Number of control actions in a state at server s*

V	*Positive constant*
V^{ij}	*Cooperative control decision*
$\widehat{V}$	*Positive number*
$\vec{V}$	*Vector of cooperative control decisions*
$\vec{V}_X$	*Vector of cooperative pairs control decisions*
$\mathscr{V}$	*Set of cooperative control decision*
W_X	*Control input*
w_n^x	*Action at terminal n in state* x_n
$\mathscr{W}_X$	*Set of control inputs*
X	*System state*
X_i	*System state at terminal i*
$\hat{X}_s$	*System state at server s*
$\mathscr{X}$	*Set of system states*
$\lvert\mathscr{X}_i\rvert$	*Number of states at terminal i*
x_n	*System state at terminal n*
$\lvert\hat{\mathscr{X}}_s\rvert$	*Number of states at serverl s*
$\vec{Y}$	*Vector of* y_i*s*
Δ	*Number of sub-slots*
α_{is}	*Non-negative parameter*
$\hat{\alpha}_s$	*Non-negative parameter*
η	*Frame index*
θ	*Positive number*
κ	*Positive number*
$\hat{\kappa}$	*Positive number*
$\tilde{\kappa}$	*Positive number*
$\vec{\lambda}$	*Average arrival rate vector*
λ_i	*Average arrival rate at terminal i*
λ_n	*Average arrival rate at terminal n*
Λ	*Stability region*
Λ_{T}	*Stability region at terminals*
Λ_{S}	*Stability region at servers*
$\vec{\mu}$	*Service rate vector*
$\vec{\mu}^*$	*Unified service rate vector*
μ_i	*Service rate at terminal i*
μ_{ij}	*Service rate from terminal i to terminal j*

μ_{nm}	*Service rate between terminal i and access point m*
μ_{is}	*Service rate from terminal i to server s*
μ_{i0}	*Requests processed at terminal i*
$\mu_{\max}^{\text{out}}$	*Maximum service rate*
$\hat{\mu}_s$	*Total service rate at server s*
$\hat{\mu}_s^i$	*Service rate server s provides to terminal i*
$\hat{\mu}^{\max}$	*Maximum supportable service rate at server s*
π	*Policy*
π_i	*Policy at terminal i*
π_n	*Policy at terminal n*
$\hat{\pi}_s$	*Policy at server s*
π_{H_i}	*Steady state probability for channel* $\vec{H}_i$
π_H	*Steady state probability for channel*
π_{H_e}	*Steady state probability for channel* $\vec{H}_e$
$\pi_{H_i}^{\text{e}}$	*Steady state probability for channel* $\vec{H}_i^e$
Π	*Set of feasible power allocation policies*
Π_i	*Set of feasible power allocation policies for terminal i*
Π_n	*Set of feasible power allocation policies for terminal n*
$\hat{\Pi}_s$	*Set of feasible power allocation policies at server s*
$\vec{\varphi}$	*Vector of binary variables*
φ_i	*Binary variable*
τ	*Time index*
ρ	*Weight*
ρ_i	*Positive number*
$\widehat{\Gamma}$	*Set of all full power long-term service rates at servers*
Γ^*	*Set of all full power long-term service rates at terminals*
AP	*Access point*
DNA	*Dynamic network architecture*
CN	*Cognitive network*
COP	*Close to optimal*
CPU	*Central processing unit*
CR	*Cognitive router*
CSI	*Channel state information*
CWN	*Conventional wireless network*
DFS	*Dynamic frequency scaling*

DFS	*Dynamic voltage scaling*
ICT	*Information and communication technology*
InTeNet	*Inter technology Networking*
IT	*Information technology*
MCC	*Mobile cloud computing*
MDP	*Markov decision process*
MIMO	*Multiple-input multiple-output*
MLI	*Minimun load index*
NC	*Non-cooperative*
NW	*Entire network*
OP	*Optimal*
PC	*Partial cognitive*
PSP	*Primary service provider*
PU	*Primary user*
QoS	*Quality of service*
QSI	*Queue state information*
RNL	*Relative network load*
RVM	*Running variance metric*
S	*Servers*
SO1	*Suboptimal policy 1*
SO2	*Suboptimal policy 2*
SSP	*Secondary service provider*
SU	*Secondary user*
T	*Terminals*
TDMA	*Time division multiple access*
TS	*Terminals to servers*
UMDP	*Unconstrained Markov decision process*
UP	*Unconstrained problem*
U1	*Terminal 1*
U2	*Terminal 2*
VDC	*Virtualized data center*
VIA	*Value iteration algorithm*
VM	*Virtual machine*
VMM	*Virtual machine monitor*
WLAN	*Wireless local area network*

Contents

1 Introduction

Efficient resource allocation schemes are one of the key elements to support the ever increasing need for high data rate services in wireless networks. There is already a vast variety of research on resource allocation for wireless networks [1]-[8]. However, we note that the many resource allocation strategies implemented so far do not consider the effects of queueing with randomly arriving traffic and time varying channels. This is despite the fact that control decisions based on both channel state information (CSI) and queue state information (QSI) have been shown to be effective in providing higher throughput and smaller delay in the presence of time varying channels and resource demands [9]-[14]. Managing queue backlog is important in providing resource allocation strategies that maximize the throughput and minimize the delay in the presence of time varying channels and changing user demands. This is of particular importance for the new emerging network paradigms like cognitive networks, dynamic network architecture (DNA) networks, cooperative networks as well as the networks with data centers and computing clouds.

In order to be able to respond to the increasing demand of high data rates, adaptation to the changes in network conditions is necessary. In this thesis, we develop dynamic network stabilizing algorithms for these advanced, new emerging wireless networks with queues and uncertain channels. The channel uncertainty is either due to fading in conventional wireless networks (CWNs) or due to uncertain channel availability both in primary service provider (PSP) and secondary service provider (SSP) cognitive networks. We define a unified control problem, where the goal is to maximize the resource utilization and at the same time provide bound for the average delay in the presence of both CWNs and PSP/SSP cognitive networks. Dynamic programming methods that utilize the availability of accurate CSI and QSI are used to provide a dynamic control policy that by adapting to the changes in network conditions stabilizes the network and optimizes the usage of the available resources for both CWNs as well as for PSP/SSP cognitive networks. While dynamic algorithms require high complexity, they provide the best possible result. Thus, the optimal control policy can be used as a benchmark for the development of more practical schemes like approximate dynamic programming based algorithms [15], [16].

In this thesis, a comprehensive stability analysis is applied to the new network paradigms that include DNA networks and distributed virtualized data centers. A new unified stability analysis in different network environments for both PSP/SSP cognitive networks and for CWNs is also provided in this thesis.

Next, in this chapter, the review of the previous research on network stability is presented in 1.1. The aims and the outline of the thesis in given in Section 1.2. Finally, the author's contributions to the original publications are described in Section 1.3.

1.1 Previous research on network stability

In this section, a brief summary of the previous work done in the field of network stability is presented.

One of the first papers that introduced the concepts of queue stability and stability region is [17]. Few years later, the work in [17] was further elaborated in [18] and [19]. The authors in [18] considered the stability of a Markov chain queue length process in multi-hop radio networks and proposed a centralized maximum throughput link scheduling policy that was proven to stabilize the network by showing that the system stability region and the stability region of the policy are equal. It was also shown in [18] that, if the one step Lyapunov drift gets negative, when the queue lengths are sufficiently large, the network is stable. The one step drift in [18] was extended into a cumulative Lyapunov drift in [20]. The study in [19] is a continuation of [18]. In [19], the authors proposed a stability condition for a system with a set of Markov chain queues competing for the service of a single server. The system in [19] was defined to be stable, if the sum of the average rates at which work is entering the queues is smaller than the sum of the proportion of the time slots the queues can receive service. In addition, the one step Lyapynov drift proposed in [18] was used to analyse the stability of a longest connected queue policy in [19], where the authors discovered that a policy can be shown to be stable if the drift is bounded above. The works in [18] and [19] have been inspiring further research on network stability and the development of network stabilizing dynamic algorithms by many researchers.

The longest connected queue policy proposed in [19] was further elaborated in [22], where the unicast scheduling problem in [19] was extended into a multicast case. The work in [22] uses the one step Lyapunov techniques to analyse the stability of scheduling and coding strategies for embedded Markov chain queues. In [23], the square of the workload was used to analyse the stability of Jackson networks [24]. The ideas in [23]

were further elaborated in [25], where they proposed a new technique for obtaining stability of Markovian queueing networks and scheduling policies. In [25] the one step quadratic Lyapunov functions were used to obtain bounds for the queue assuming that the system is stable. The work in [25] was later extended in [26], where a programmatic procedure to analyse the stability of queueing networks and scheduling policies was proposed. The goal in [26] was to programmatically guarantee a negative one step quadratic Lyapunov drift for the buffers that get too large, even for the systems that are not Markovian. More work on the one step Lyapunov drift stability analysis can be found for switch and router architectures in [27], [28] and [29].

The works in [18], [19] and [20] have inspired the research in [12], where the authors established a network stability region and develop capacity achieving power allocation and routing policies for general networks with queues, wireless links and adaptive service rates. Most of the results on stability in [12], are also published in [10], [13], [14] and [30]. The authors in [10], [12], [13] extend the cumulative Lyapunov drift theorem in [20] into a K-step Lyapunov drift and use it to provide new bound for the average queue length. In addition, in order to analyse the stability of their dynamic routing and power control policy (DRPC), stationary resource allocation policy were proposed in [10], [12]. By using the K-step Lyapunov drift theorem, the stationary policy was used to provide bound for the average queue length in [10], [12]. As the performance of the DRPC is expected to be better than the performance of the stationary policy, the proposed bound was assumed to be valid also for DRPC.

A methodology called Lyapunov optimization was proposed in [14], [30] for designing control policies to maximize long-term average utility subject to the network stability. Just like in [10], [12] and [13], the works in [14], [30] uses the negativity of the K-step Lyapunov drift theorem to provide bound for the average queue. Lyapunov optimization has been used to guarantee network stability optimal cross-layer control policies for wireless networks also in [31], [32] and [9]. The work on stability and the bound provided by the extended Lyapunov drift theorem in [12] has been used to analyse the stability of different resource allocation policies also in [9], [31]-[34].

1.2 Outline and contributions of the thesis

The goal of this thesis is to present a number of new paradigms in wireless networks with queues and time varying channels. For such networks, the contributions include solutions for optimal and sub-optimal dynamic resource allocation, topology control and

network stability. Unified system models, unified control problems, unified network stability regions and unified stability analysis are provided for both SSP and PSP cognitive networks as well as for CWN. Novel approaches to the stability analysis of dynamic control policies in wireless networks are also provided in this thesis.

The thesis is organized into 6 chapters:

– Introduction and the literature review of the previous work on network stability is presented in Chapter 1.
– The concept of network stability and the analytical tools used to analyse the stability of time varying queueing networks are presented in Chapter 2.
– The content of Chapter 3 have been presented in [21], where a virtualized data center (VDC) is considered. The VDC consists of a set of servers hosting a number of mobile terminals forming a mobile cloud. Using this model, a unified dynamic optimization problem is formulated for both PSP and SSP cognitive networks and for CWN. The goal is to maximize the joint utility of the long-term application processing throughput of the terminals and to minimize the average total power usage while keeping the network stable. Dynamic programming methods are used to provide a new unified dynamic control policy for PSP and SSP cognitive networks as well as for CWNs.

  A unified stability region is illustrated and a new unified stability analysis is proposed for both CWNs and PSP/SSP cognitive networks. The K-step Lyapunov drift theorem is used to analyse the stability of the proposed optimal control policy. Our policy is shown to outperform the stationary and the frame based policy proposed in [10],[12] and to stabilize the network.

  Numerical results are provided to support our stability analysis and to evaluate the performance of the dynamic control policy in the presence of cognitive wireless networks and CWNs.
– In Chapter 4, an optimal cooperative network control problem over time-slotted channels with uncertainties is considered. The uncertainties can be either due to fading in CWNs or due to uncertain channel availability in PSP/SSP cognitive networks. For this model, a unified optimization problem is formulated and dynamic programming tools are used to provide a unified optimal cooperative control policy for PSP and SSP cognitive networks as well as for CWNs.

  In order to compare the performance of the cooperative communication to the conventional non-cooperative case, the stability regions are illustrated for both

networks. In addition, when the control actions need to be calculated for each network state, it is important to note that the best network stabilizing policy is the one that minimizes the maximum queue length over all terminals. The K-step Lyapunov drift is used to analyse the stability of the optimal control policy and show that the proposed policy stabilizes the network and minimizes the long-term average maximum queue length.

For illustration purposes and to validate our stability analysis, the performance of the optimal dynamic control policy is evaluated with simulations. The results show that by adapting to the changes in network conditions, the cooperative policy mitigates the effects of PSP and SSP cognitive networks on each other.

– In Chapter 5, a new paradigm in wireless network access is presented and analysed. Each terminal in an ad hoc or multi-hop cellular network can be turned into an AP any time, when it is connected to internet. A new topology control policy is proposed to maximize the network performance with minimum power consumption and to stabilize the network. As the implementation of the optimal policy required full knowledge of CSI and QSI, approximate dynamic programming methods and one step VIA are used to provide new suboptimal control policies.

  In order to evaluate the performance of different policies, the network stability region and the stability regions of the optimal and sub-optimal policies are illustrated. Using the K-step Lyapunov drift, the stability and the performance of the optimal dynamic policy is analysed. Our optimal control policy is shown to stabilize the network and minimizes the bound for the average delay.

  The numerical results are provided in order to support our stability analysis and to compare the performance of the optimal and sub-optimal control policies to each other.

– Finally, the main results are summarized in Chapter 6. The future research directions on this field are also considered in Chapter 6.

The main contributions of this thesis are included in Chapters 3-5.

1.3 Author's contributions to the thesis

The work on this thesis is based on publications [21], [35] and additional research. The journal paper [21] has been coauthored with Prof. Savo Glisic, Prof. Yuguang Fang and Dr. Pan Li. The conference paper [35] has been coauthored with Prof. Savo Glisic. Author's supervisor Prof. Savo Glisic provided reviews and suggestions related

to technical issues, editorial corrections and publication process. Prof. Yuguang Fang has reviewed journal [21], and Dr. Pan Li has provided some comments on [21]. The author had the main responsibility for providing the results, analysis and writing the papers and the thesis.

2 Analytical tools for stability analysis of time varying queueing networks

In this chapter, we first present the definition for network stability following with the most important tools used to analyse the performance and stability of different resource allocation policies in wireless networks.

2.1 Network stability

Consider a wireless network with a set of $\mathscr{I}$ terminals with queues. Let i denote the index of a terminal and $i = \{1,2,....,|\mathscr{I}|\}$, where $|\mathscr{I}|$ denotes a number of terminals within the set $\mathscr{I}$. We use $q_i(t)$ to represent the amount of packets (each of size b bits) in the queue of terminal i in time slot t and let $\vec{Q}(t) = [q_1(t),...,q_{|\mathscr{I}|}(t)]$ denote the vector of such queue lengths. These queues evolve according to time varying arrival processes $a_i(t)$ and service rates $\mu_i(t)$ as

$$q_i(t+1) = \max\{q_i(t) - \mu_i(t), 0\} + a_i(t), \tag{1}$$

where $a_i(t)$ represents the number of arriving packets in time slot t and $\mu_i(t)$ is the number of packets that can be released in slot t.

As a measure of the fraction of time the unfinished work in the queue is above a certain value $\hat{V}$, an overflow function $\underline{g}(\hat{V})$ is presented as

$$\underline{g}(\hat{V}) = \limsup_{t\to\infty} \frac{1}{t}\mathbb{E}\Big\{\sum_{t=0}^{t} 1_{[q_i(t)>\hat{V}]}\Big\}, \tag{2}$$

where $\mathbb{E}\{\cdot\}$ denotes the expected value and the indicator function $1_{[X]}$ takes the value 1 whenever X is satisfied, and 0 otherwise [12]. A single queue is stable, if $\underline{g}(\hat{V}) \to 0$ as $\hat{V} \to \infty$ [12]. The whole network is stable, if all individual queues are stable [12].

2.2 Network stability region

Let λ_i represent long-term average arrival rates for terminal i given as

$$\lambda_i = \lim_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} a_i(\tau) \tag{3}$$

In addition, let $\vec{\lambda} = [\lambda_1, \lambda_2, ..., \lambda_{|\mathscr{I}|}]$ denote the vector of these arrival rates.

Network stability/capacity region Λ is the set of all long-term average arrival rates that the network can stably support considering all the resource allocation policies that we can have for the network [18]. Correspondingly, stability region of a specific resource allocation policy is a set of all long-term arrival rates that a policy can stably support and it is a subset of the network stability region [18]. For arrival rates outside the network stability region, the network cannot be stabilized [18].

If a policy supports every point on the network stability region, it is called throughput optimal. Then, the network stability region and the stability region of the policy are equal. Thus, the network stability region can be used as an important criteria to analyse the performance of different resource allocation policies, i.e., the larger the stability region of the policy, the better.

Let $\vec{\mu} = [\mu_1, \mu_2, ..., \mu_{|\mathscr{I}|}]$ denote the vector of service rates, $\underline{H}$ channel state, $\pi_{\underline{H}}$ steady state probability for channel $\underline{H}$, $\mathscr{H}$ the set of channel states, $\underline{P}$ a power allocation policy and Π the set of the feasible power allocation policies. In addition, let g_i represent long-term average supportable service rate at terminal i and $\vec{G}$ the vector of these average service rates. Due to the time varying channel state conditions, $\vec{G}$ must be averaged over all possible channel states $\underline{H} \in \mathscr{H}$. Moreover, $\vec{G}$ is not fixed and depends on resource allocation policy for choosing the control actions for each $\mathscr{H}$. The network stability region, i.e, the set of all long-term average service rates $\vec{G}$ that the network can be configured to support, can then be given as [12]:

$$\sum_{\underline{H} \in \mathscr{H}} \pi_{\underline{H}} \text{Conv}\{\underline{P}, \underline{H}) | \underline{P} \in \Pi\}, \tag{4}$$

where $\text{Conv}\{\Upsilon_{\underline{H}}\}$ represents convex hull of the set $\Upsilon_{\underline{H}}$ that is defined as the set of all convex combinations $p_1 b_1 + p_2 b_2 + ... + p_j b_j$ of elements $b_j \in \Upsilon_{\underline{H}}$ and p_js are probabilities summing up to 1. Specifically, the throughput region in (4) can be viewed as a set of all long-term average service rates that the network can be configured to support.

2.3 Lyapunov stability

Lyapunov drift has been used as one of the most important tools to analyse network stability and the stability of different resource allocation policies by many researchers [10], [12], [13], [14], [18], [19], [22] and [30].

According to the author's best knowledge, the initial forms of the drift were first used to provide a sufficient condition for stability of a queuing system in [18], [19]. The authors in [18] defined the sufficient condition for stability, where $q_i(t)$ can be considered to be stable, if

$$\mathbb{E}\{L[q_i(t+1)] - L[q_i(t)] | q_i(t)\} \leq -\hat{\kappa} \quad \forall \quad q_i(t) \geq \tilde{Y}, \tag{5}$$

where $L[q_i(t)] = q_i(t)^2$ and $\hat{\kappa} > 0$. The condition in (5) was later used to prove the stability of the longest connected queue policy in [19]. Since then, the condition in (5) has been elaborated to analyse the stability of different resource allocation policies by many researchers.

We assume that $q_i(0) < \infty$ for all terminals and define the Lyapunov drift as $L[Q(t)] = \sum_{i \in \mathscr{I}} q_i(t)^2$. The 1-step Lyapunov drift can now be given as

$$\mathbb{E}\{L[Q(t+1)] - L[Q(t)] | \vec{Q}(t)\} \leq B - 2 \sum_{i \in \mathscr{I}} q_i(t) \mathbb{E}\{\mu_i(t) - a_i(t) | \vec{Q}(t)\}, \tag{6}$$

where B is a positive constant. If there exists $\kappa > 0$ such that over all the terminals and for all time slots t

$$\mathbb{E}\{\mu_i(t) - a_i(t) | q_i(t)\} \geq \kappa, \tag{7}$$

the 1-step drift can be defined as

$$\mathbb{E}\{L[Q(t+1)] - L[Q(t)] | \vec{Q}(t)\} \leq B - 2\kappa \sum_{i \in \mathscr{I}} q_i(t). \tag{8}$$

Taking expectations of the above inequality over the distribution of queue length, summing over time slots, dividing by t and using the non-negativity of the Lyapunov function, we see that the network is stable, and the bound for the average queue length is given as

$$\limsup_{t \to \infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \sum_{i \in \mathscr{I}} \mathbb{E}\{q_i(t)\} \leq \frac{B}{2\kappa}. \tag{9}$$

When the network stochastics require more than one time slot to ensure a negative drift, the 1-step Lyapunov drift in (6) must be extended into a K-step Lyapunov drift [20] , where the one step drift proposed in [18] is evaluated K steps into the future. The K-step drift has been used to analyse the stability of different algorithms in several publications, such as [9], [12], [13], [30]-[34] and [36]-[40].

If the Lyapunov function is defined as $L[Q(t)] = \sum_{i\in\mathscr{I}} q_i^2(t)$, the K-step Lyapunov drift is given as

$$\mathbb{E}\{L[Q(K+t)] - L[Q(t)]|\vec{Q}(t)\} \leq K^2|\mathscr{I}|V - 2\sum_{i\in\mathscr{I}} q_i(t) \sum_{\tau=t}^{t+K-1} \mathbb{E}\{\mu_i(\tau) - a_i(\tau)|\vec{Q}(t)\}. \tag{10}$$

where V is a positive constant and $t \in \{0,1,...,K-1\}$.

Consider (10) at times $t = \hat{j}K + t_0$, and by summing over t_0 and $\hat{j}$ from 0 to $K-1$ and from 0 to $\hat{J}-1$, we get

$$\sum_{t_0=0}^{K-1}\sum_{\hat{j}=0}^{\hat{J}-1} \mathbb{E}\{L[Q((j+1)K+t_0)] - L[Q(jK+t_0)]|\vec{Q}(jK+t_0)\} \leq \tag{11}$$
$$K^3|\mathscr{I}|V\hat{J} - 2\sum_{t_0=0}^{K-1}\sum_{\hat{j}=0}^{\hat{J}-1}\sum_{i\in\mathscr{I}} q_i(jK+t_0) \sum_{\tau=jK+t_0}^{(j+1)K+t_0-1} \mathbb{E}\{\mu_i(\tau) - a_i(\tau)|\vec{Q}(jK+t_0)\}.$$

If there exists a positive constant $\tilde{\kappa} > 0$ such that over all terminals and for each time slot t

$$\sum_{\tau=jK+t_0}^{(j+1)K+t_0-1} \mathbb{E}\{\mu_i(\tau) - a_i(\tau)|q_i(jK+t_0)\} \geq \tilde{\kappa}, \tag{12}$$

(11) can be rewritten as

$$\sum_{t_0=0}^{K-1}\sum_{\hat{j}=0}^{\hat{J}-1} \mathbb{E}\{L[Q((j+1)K+t_0)] - L[Q(jK+t_0)]|\vec{Q}(jK+t_0)\} \leq \tag{13}$$
$$K^3|\mathscr{I}|V\hat{J} - 2\tilde{\kappa}\sum_{t_0=0}^{K-1}\sum_{\hat{j}=0}^{\hat{J}-1}\sum_{i\in\mathscr{I}} q_i(jK+t_0).$$

Taking expectations of the above inequality, dividing by $\hat{J}K$ and using the non-negativity of the Lyapunov function, we see that the network is stable, and the bound for the average queue length is given as

$$\limsup_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1}\sum_{i\in\mathscr{I}} \mathbb{E}\{q_i(\tau)\} \leq \frac{K^2|\mathscr{I}|V}{2\tilde{\kappa}}. \tag{14}$$

3 Resource harvesting in cognitive wireless computing networks with mobile clouds and virtualized distributed data centers: performance limits

In this chapter, we consider a virtualized date center (VDC) consisting of a set of servers hosting a number of mobile terminals forming a mobile cloud, and study the problem of resource allocation in the presence of time varying workloads and uncertain channels. The channel uncertainty may be either due to fading in conventional wireless networks (CWNs) and/or uncertain link availability and reliability both in primary service provider (PSP) and secondary service provide (SSP) cognitive wireless networks.

First, a detailed discussion on developing a unified system model for the VDC (computing cloud) for both PSP and SSP cognitive networks as well as for CWNs is presented. For this unified model, we characterize the joint stability region and propose a new unified stability analysis for both PSP and SSP cognitive networks (CNs) and for CWNs.

We formulate a unified control problem, where the goal is to maximize a joint utility of the long-term application processing throughput of the terminals and to minimize the average total power usage in the overall system while keeping the network stable. As the control problem is to dynamically adjust resources according to channel and workload fluctuations, we reformulate the problem into a Markov Decision Process (MDP) and use dynamic programming and Value Iteration algorithm (VIA) [41], [42] to provide a dynamic control policy that solves the problem for PSP and SSP cognitive networks and for CWNs. We also analyse the complexity of the proposed dynamic control algorithm.

Lyapunov drift theorem is used to analyse the stability of our dynamic control policy. It is shown that the policy supports every point on the network stability region and outperforms the stationary control policy presented in [10]. A frame based policy was proposed to stabilize the network and to outperform the stationary policy in [10], [12]. In this chapter, we show that the performance of the frame based policy is not better than the performance of the stationary randomized policy and that the frame based policy does not provide bound for the average queue length, when there are shared resources in the network.

Finally, the performance of the dynamic policy is illustrated with simulations in the presence of time varying workloads and uncertain channels for both CWNs and cognitive wireless networks. The simulations support our stability analysis presented in chapters 3.5 and 3.7. It is also shown by simulations that by adapting to the changes in network conditions, our control policy mitigates the effect of PSP and SSP cognitive networks on each other.

The remainder of this chapter is organized as follows. Motivation and the related work are presented in Section 3.1. Section 3.2 describes the system model and Section 3.3 presents the optimization problem formulation. In Section 3.4, we reformulate the problem as a MDP, and propose the dynamic control policy. The joint network stability regions for both SSP and PSP cognitive networks and also for CWNs are derived in Section 3.5. The complexity of the dynamic policy is analyzed in Section 3.6. In Section 3.7, the unified stability analysis for both PSP and SSP cognitive networks and for CWN is introduced. The simulations are conducted to validate the theoretical analysis of this chapter and are presented in Section 3.8. Finally, some concluding remarks are offered in Section 3.9.

3.1 Motivation and related work

In this section, the motivation behind the research and the related work in this area are presented.

3.1.1 Motivation

Cloud computing [43]-[46] has gained a lot of attention in recent years as the next generation computing infrastructure [47]-[51]. There are many surveys that recognize the importance and miscellaneous usability of cloud computing in the near future [52]-[57]. Cloud computing allows users to use resources such as servers, storages, platforms and applications provided by cloud providers remotely and at low cost over the Internet [58]. Cloud computing is a great technology for its users as it can not only satisfy even the most intensive computing needs of the customers but it also releases them from IT infrastructure investments, the complexity of IT management, planning and maintenance.

Another key advantage of cloud computing is its dynamic provision of computing resources and services. The cloud resources can be rapidly provisioned and released with

minimal management efforts and service provider interaction [59]. As a result, cloud computing resources can be dynamically shared by multiple users in a pay-as-you-go fashion and based on users' resource demand. Since these capabilities are provided at relatively low costs compared to several individual fixed infrastructures managed by a company or user, cloud computing provides a promising solution for environmentally conscious IT [54], [60], [61]. There are already a number of organizations that host and/or offering cloud computing services. At the moment the leading cloud service providers are Google [62], Amazon [63] and IBM [64].

Cloud computing has also been proposed as an answer to ever increasing demand for computing resources of the mobile devices by many researchers [52], [53], [65]. Nowadays, smart phones can be connected to Internet either through telecom network or access points using Wi-Fi [53]. The phones are also capable of supporting a wide range of applications that significantly increase their demand on storage, computing resources and power. However, due to the obstacles such as low processing power, limited battery life, storage, memory and bandwidth, several applications are still unsuitable for smart phones [66]. By integrating cloud computing with the mobile devices, mobile cloud computing (MCC) [59] has been introduced as a potential technology to overcome the resource restrictions on mobile computing [52] allowing mobile users to achieve a wide variety of mobile services at low cost.

In MCC, mobile devices' resource intensive computations or data storage happen remotely at a resource rich cloud. The transfer of computationally intensive applicatios to be performed at a remote server is commonly referred as computational offloading. It has been shown in [67] that offloading can save battery life in mobile phones and decrease the execution times of applications. Applications that can benefit from computational offloading related to image processing and games, for example, can be found in [68] and the references therein. Centralized monitoring and maintenance of the software also increases the security level for mobile devices [65]. With the MCC, more environmental friendly mobile computing and the dream of information anytime and anywhere is becoming reality. In addition, by reducing the development and running costs of the mobile devices and applications, MCC has also gained the attention of entrepreneurs as a promising provider of profitable business options and new technical functionalities.

3.1.2 Related work

The issue of power consumption in the information and communication technology (ICT) sector has been receiving increasing attention in recent years [54], [65], [69], [70]. As cloud computing services become increasingly popular among both businesses and private users, there is increasing demand for the computational services of the high performance large-scale data centers, which consume enormous amounts of electrical power. There is therefore a growing interest in improving the energy efficiency of today's data centers and cloud computing facilities [52]. Since the data centers' energy consumption have huge impact on environment, providing energy efficient, desired quality of service (QoS) or service level objectives satisfying, resource management systems for cloud service data center are increasingly important.

Unfortunately, resources inside the data centers often operate at low utilization due to inefficient resource allocation [71]. For example, a single idle server can draw as much as 65% of peak power value if not turned off [72]. In current systems, servers are also under-used most of the time, as applications' resource demands are easily over-estimated in order to handle even the most demanding workloads. As a result, applications hold resources that they hardly need at all, since large workloads may be rare. Ideally, unused resources should be released for other applications to use. Efficient resource management schemes are the key to maximizing the use of the resources both at the cloud service users and at the data center.

Data center virtualization has been shown to offer great benefits in reducing the total power consumption and increasing reliability allowing multiple heterogeneous applications to share resources and run simultaneously on a single server [73], [74], [75]. Virtual Machine (VM) technology increases server utilization by enabling consolidation of multiple applications on the same server and the sharing of resources among these applications. By using this technology, it is possible to control the data center so that the VMs occupy only the necessary resources to serve their applications. However, achieving the right balance between consolidation and resource utilization of each application is a critical issue for applications with time-varying demands.

Since workload adaptive resource allocation is important to create high performance data centers, dynamic resource allocation in VDCs has been a hot topic among researchers [71], [76], [77]. In [71], [76] and [77] feedback-driven resource control systems are designed to automatically adapt to dynamic workload changes and to meet service level objectives of applications within the shared virtualized infrastructure. Such

techniques use a feedback control loop, where the goal is to allocate resources to meet its performance target. However, since feedback techniques require information about the target performance level, they cannot be used when the goal is to maximize utility. In [78], the authors propose a dynamic live placement scheme for applications in cloud computing environments called EnaCloud, where an energy-aware heuristic algorithm is proposed to minimize the number of running VMs. Much of the previous work on resource allocation in the VDCs is based on proactive workload adaptive resource provisioning and steady state queuing models [79], [80], [81]. The work in [79] defines a dynamic resource provisioning problem for virtualized server systems as a sequential optimization problem which is solved using a lookahead control. Such a technique is quite useful when control actions have deadlines to meet, but requires estimates of future workloads. In [81], dynamic resource provisioning in a virtualized service environment is based on the estimate of the power usage behavior of the hosted applications. Three online workload adaptive resource control mechanisms based on steady state queueing analysis, feedback control theory and the combination of these two are proposed in [80]. This approach requires implementation of the statistical models for the workload, and resource allocation decisions are then made to meet such a predicted resource demand. When predictions are accurate, proactive resource allocation does provide very good performance [82]. In practice, however, predictions may be inaccurate and expensive since they require workload data analysis and storage space. Research, closest to our work, can be found in [9]. The work in [9] uses Lyapunov optimization [14] to design an online control, routing and resource allocation algorithm for a VDC. While this algorithm adjusts to workload fluctuations, it does not take into account the possible channel variations between the terminals and the servers. By considering the changing user demands, control decisions based on both the channel variations and the workload, have been shown to be effective in providing higher throughput and smaller delay in the presence of time varying channels and resource demands [11], [12].

In this chapter, a new dynamic resource allocation policy for virtualized cloud service data center in the presence of uncertain channels and time varying workloads is proposed. The channel uncertainty is either due to fading in CWNs and/or uncertain link availability in PSP/SSP cognitive wireless network. By adapting to the changes in network conditions, the proposed dynamic policy maximizes the long-term application processing throughput of the terminals and minimize the average total power usage in the overall system while guaranteeing the network stability. Our dynamic control policy is shown to support every point on the network stability region, and has been proven to

be stable using the Lyapunov drift theory. In [10] and [12], a randomized stationary policy and a frame based algorithm were used to analyse the stability of a dynamic algorithm. It is shown in [10], [12] that the performance of their dynamic algorithm is fixed amount worse than the performance of the randomized stationary and the frame based algorithms. In this chapter, we prove that the performance of our dynamic policy is better than the performance of the stationary policy and propose a new unified stability analysis for both PSP and SSP cognitive networks as well as for CWNs. Different from the works that use steady state queuing and channel models, our approach makes use of both the queue length state information (QSI) and the channel state information (CSI) to dynamically adjust the available resources to meet the demand, and to increase the reliability and resource utilization of the data center.

3.2 System model and assumptions

We consider a network composed of a VDC and a number of mobile terminals with queues belonging to different clusters of mobile clouds. Let $\mathscr{I}$ denote the set of terminals within a cloud and the VDC is composed of a set of servers $\mathscr{S}$ hosting the cloud, as illustrated in Fig. 1. The servers are processing certain applications delegated to them by the terminals, for either energy saving or due to the lack of necessary software at the terminal to process the applications and the VDC may be either centralized or distributed across the network as in the network with caching [83]. However on purpose, we do not want to limit our work on a specific network architecture. Our analysis is valid for any data center with partitioning (virtualization) of the processing resources (centralized or distributed) and any conventional or PSP/SSP cognitive network characterized by the primary user (PU) return probability and secondary user (SU) channel sampling quality. By definition, mobile cloud is a set/cluster of terminals that share a certain pool of resources [84]. In our case, the terminals share the resources located at the data center.

In order to increase the energy efficiency of cognitive networks, the concepts of SSP and PSP cognitive networks have been recently introduced in [85]. In this concept, SSP provides channel state information for secondary users (SUs) so that the complexity is allocated to the network rather than to the terminals. In this way, a wide range of terminals can operate as SUs and terminals do not need to have cognitive capabilities.

Let $|\mathscr{S}|$ denote the number of servers within the data center and $|\mathscr{I}|$ represent the number of terminals within the cloud. Each server s is transformed into $|\mathscr{I}|$ VMs, each capable of serving a terminal. For simplicity, we assume that each mobile terminal can

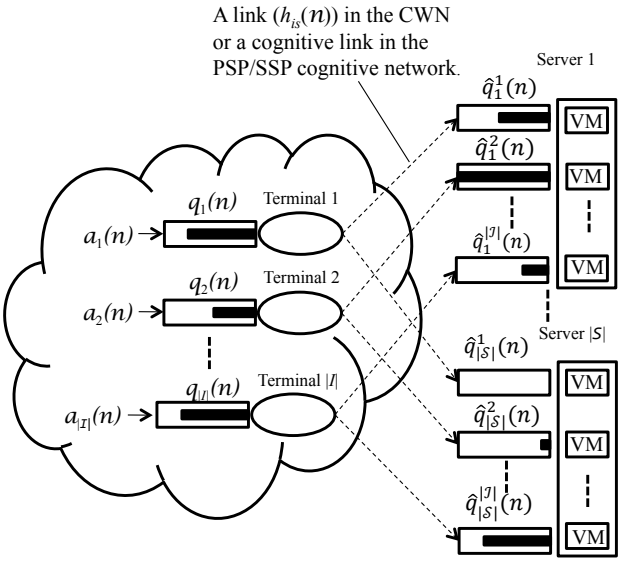

Fig. 1. A cloud with a VDC.

request service only from one server at a time, but the hosting server can change in time. By dividing the time into frames with index n, we define the following parameter for each terminal i and server s:

$$b_{is}(n) = \begin{cases} 1; & \text{If terminal } i \text{ is served on a VM of server } s \text{ in frame } n. \\ 0; & \text{Otherwise.} \end{cases}$$

Let $\vec{B}_i(n) = [b_{i1}(n), ..., b_{i|\mathcal{S}|}(n)]$ denote the vector of these parameters in frame n.

Application requests arrive to each terminal i according to a process $a_i(n)$ at the beginning of each frame n. The arrival processes $a_i(n)$ are stationary and ergodic with average rates λ_i requests/frame. The external arrivals $a_i(n)$ are bounded in their second moments every frame and $\mathbb{E}\{[a_i(n)]^2\} \leq (a^{\max})^2$ for all $i \in \mathcal{I}$. Let $\vec{A}(n) = [a_1(n), ..., a_{|\mathcal{I}|}(n)]$ denote the vector of these arrivals. For analysis purpose, we assume that the application requests are placed into infinite length transmission buffers $q_i(n)$, that are later defined in Subsection 3.2.3.

3.2.1 Channel model

Let $|h_{is}(n)|^2$ represent the channel gain between terminal i and server s. A block fading model is assumed so that the channel values remain fixed during a frame and may change from frame to frame according to a Markov chain [1]. Let $\vec{H}_i(n) = [|h_{i1}(n)|^2, |h_{i2}(n)|^2, ..., |h_{i|\mathscr{S}|}(n)|^2] \in \mathscr{H}_i$ denote the vector of channel gain processes at terminal i in frame n. The channel process $\vec{H}_i(n)$ is stationary and ergodic and takes values on a finite state space $\mathscr{H}_i$. Since the servers can have different locations, it is possible that the channels between terminal i and different servers are different.

If the channel is used within the CWN, the channel gain vector is given by $\vec{H}_i(n)$ in every frame n. Let π_{H_i} represent the steady state probability for the channel state $\vec{H}_i$ in the CWN. The channel processes are channel convergent with steady state probabilities π_{H_i}.

If the channel is used within the cognitive network, the equivalent channel gain process $\vec{H}_i^e(n)$ will have the following form:

$$\vec{H}_i^e(n) = \begin{cases} \vec{H}_i(n); & \text{With probability } p_H^{\mathrm{P}} \text{ for PSP CN or} \\ & \text{with probability } p_H^{\mathrm{S}} \text{ for SSP CN.} \\ 0; & \text{With probability } p_0^{\mathrm{P}} \text{ for PSP CN or} \\ & \text{with probability } p_0^{\mathrm{S}} \text{ for SSP CN.} \end{cases}$$

For the PSP cognitive network,

$$p_H^{\mathrm{P}} = (1 - p_1^{\mathrm{S}}) + p_1^{\mathrm{S}} p_{\mathrm{pd}} \tag{15}$$

and

$$p_0^{\mathrm{P}} = p_1^{\mathrm{S}}(1 - p_{\mathrm{pd}}). \tag{16}$$

We assume that PU transmits a preamble prior to message transmission to clear the channel in case that SU is using it (with probability p_1^{S}). Secondary user detects correctly that preamble and clears the channel with probability p_{pd}. Let p_1^{P} represent the probability that a PU is active and p_{id} is the probability that a SU detects the idling

[1] The finite state block fading Markov chain has been widely used to model the channel in the literature, e.g. [12], [15], [20]. The model has been used to mathematically characterize Rayleigh fading channel in [86] and [87]. Using block fading model for the channel, we can dynamically generate artificial channel states that are analytically tractable and can provide closed-form results. The assumption that the channels hold their states during a frame is approximation that is valid for systems whose frames are short in comparison to the channel variation. In practice, channels may vary continuously.

channel. The derivation of the probability 1 - p_1^{P} is given in [88]. In the SSP cognitive network, p_H^{S} is then given as

$$p_H^{\mathrm{S}} = (1 - p_1^{\mathrm{P}}) p_{\mathrm{id}} \tag{17}$$

and the probability that the channel cannot be used is

$$p_0^{\mathrm{S}} = (1 - p_1^{\mathrm{P}})(1 - p_{\mathrm{id}}) + p_1^{\mathrm{P}}. \tag{18}$$

In other words, SU gets the channel $\vec{H}_i(n)$, if the PU is not active and the SU detects the idling channel. The channel is not used, if PU is not active but the SU fails to detect the idling channel or the PU is active. Let $\pi^e_{H_i}$ denote the steady state probability for channel state $\vec{H}_i^e$ in PSP/SSP cognitive networks given as

$$\pi^e_{H_i} = \begin{cases} p_H^{\mathrm{P}} \pi_{H_i} / p_H^{\mathrm{S}} \pi_{H_i}; & \text{When } \vec{H}_i^e = \vec{H}_i. \\ 1 - p_H^{\mathrm{P}} / 1 - p_H^{\mathrm{S}}; & \text{When } \vec{H}_i^e = 0. \end{cases}$$

We use $I(n)$ to denote the channel availability indicator at the beginning of a frame n. For the SSP cognitive network, $I(n)$ is defined as

$$I(n) = \begin{cases} 1; & \text{If } \vec{H}_i^e(n) = \vec{H}_i(n). \\ 0; & \text{If } \vec{H}_i^e(n) = 0. \end{cases}$$

The probability that $I(n) = 1$ is $p[I(n) = 1] = p_H^{\mathrm{S}}$ and the probability that $I(n) = 0$ is $p[I(n) = 0] = p_0^{\mathrm{S}}$. For the PSP cognitive network, $I(n)$ is given as

$$I(n) = \begin{cases} 1; & \text{If } \vec{H}_i^e(n) = \vec{H}_i(n). \\ 0; & \text{If } \vec{H}_i^e(n) = 0. \end{cases}$$

and the probabilities are $p[I(n) = 1] = p_H^{\mathrm{P}}$ and $p[I(n) = 0] = p_0^{\mathrm{P}}$.

In addition, for the given channel in the SSP cognitive network, we define a channel corruption indicator $I_{\mathrm{r}}(n)$ during a frame n. In the SSP cognitive network, $I_{\mathrm{r}}(n)$ is given as

$$I_{\mathrm{r}}(n) = \begin{cases} 0; & \text{If PU returns to the channel.} \\ 1; & \text{Otherwise.} \end{cases}$$

where probabilities $p[I_{\mathrm{r}}(n) = 1] = 1 - p_{\mathrm{return}}^{\mathrm{P}}$ and $p[I_{\mathrm{r}}(n) = 0] = p_{\mathrm{return}}^{\mathrm{P}}$. The PU return probability $p_{\mathrm{return}}^{\mathrm{P}}$ is discussed in [88]. The channel corruption indicator $I_{\mathrm{r}}(n)$ in the PSP cognitive network is given as

$$I_{\mathrm{r}}(n) = \begin{cases} 0; & \text{If SU returns to the channel and does not} \\ & \text{detect the presence of PU (collision).} \\ 1; & \text{Otherwise.} \end{cases}$$

The probabilities are given as

$$p[I_{\mathrm{r}}(n)=1]=(1-p^{\mathrm{S}}_{\mathrm{return}})+p^{\mathrm{S}}_{\mathrm{return}}p_{\mathrm{sd}} \tag{19}$$

and

$$p[I_{\mathrm{r}}(n)=0]=p^{\mathrm{S}}_{\mathrm{return}}(1-p_{\mathrm{sd}}), \tag{20}$$

where $p^{\mathrm{S}}_{\mathrm{return}}$ is the probability of SU returning to the channel and p_{sd} is the probability that SU correctly detects the presence of PU.

Additional modification of the channel model includes the option what we refer to as "partial cognitive networks" (PC networks), where the network operator's overall resources include both cognitive and conventional (purchased) links [85]. Given π_{H_i}, $\pi^{e}_{H_i}$, $\vec{H}^{e}_{i}$ and $\vec{H}_i \in \mathscr{H}_i$, deriving the channel model for the PC network is straightforward.

3.2.2 Power consumption

Depending on the current workloads, current channel states, available energy and needed software, the application requests can be processed either at the terminal or delegated to be performed at one of the servers hosting the terminal. Let $\mu_{is}(n)$ denote the number of requests delivered from terminal i to be processed at the hosting server s in frame n. Let $\mu_i(n)$ represent the number of requests processed at terminal i in frame n, when there is a channel available between terminal i and server s, i.e., $\vec{H}_i(n) \in \mathscr{H}_i$. In addition, let $\mu_{i0}(n)$ denote the number of requests that can be processed at terminal i only, when there is no channel available between terminal i and server s in frame n, i.e., $I(n)=0$. When $I(n)=0$, more applications might be processed at terminal i only and $\mu_{i0}(n) \geq \mu_i(n)$.

We use $P^{\mathrm{tot}}_{i}(n)=P_i(n)+P_{is}(n)$ to represent the total power consumption of terminal i in frame n, where $P_i(n)$ is the power required to process application requests at terminal i and $P_{is}(n)$ is the power required to deliver requests to be processed at server s. Let α_i and α_{is} denote non-negative parameters. In the CWN, we have

$$P_i(n)=\mu_i(n)\alpha_i \tag{21}$$

$$P_{is}(n)=\frac{\mu_{is}(n)\alpha_{is}}{|h_{is}(n)|^2}. \tag{22}$$

In the PSP/SSP cognitive networks, $P_i(n)$ and $P_{is}(n)$ are given as

$$P_i(n)=I(n)\mu_i(n)\alpha_i+(1-I(n))\mu_{i0}(n)\alpha_i \tag{23}$$

$$P_{is}(n) = \frac{I(n)\mu_{is}(n)\alpha_{is}}{|h_{is}(n)|^2}. \tag{24}$$

Let $P^{\max}$ denote the maximum power available at each terminal in frame n.

Each server s has a set of resources that are allocated to the VMs hosted on it by its resource controller. These resources can include, for example, the data center power and the necessary software at the data center that is not available at the terminals. Both of these resources can be easily added into the system model, as described in Sections 3.3 and 3.4. However, in this chapter, we only focus on the CPU frequency and power constraints. All servers are assumed to have identical CPU resources. In our model, CPUs run at finite number of operating frequencies $f_{\min} < f < ... < f_{\max}$. At each utilization level f, the power consumption at server s is estimated as $\hat{P}_s(f) = \hat{P}_{\min} + \theta(f - f_{\min})^2$ [9]. Available techniques such as dynamic frequency scaling (DFS), dynamic voltage scaling (DVS) and combination of the two can be used to change the current CPU frequency that affects the CPU power consumption [89], [90]. The maximum power at server s is given as $\hat{P}^{\max} = \hat{P}_{\min} + \theta(f_{\max} - f_{\min})^2$. At utilization level f the maximum supportable service rate $\hat{\mu}^{\max}(f)$ at server s is given as [9]

$$\hat{\mu}^{\max}(f) = \frac{\hat{P}_s(f)}{\hat{\alpha}_s} = \frac{\hat{P}_{\min} + \theta(f - f_{\min})^2}{\hat{\alpha}_s}, \tag{25}$$

where $\hat{\alpha}_s$ represents a non-negative parameter. The VM's resource allocation can be changed dynamically online without disrupting the running applications within the VMs [91]. The resources for each VM are adapted to the changing workloads during its lifetime. In virtualized server environment the virtual machine monitor (VMM) at any physical machine handles resource multiplexing and isolation between VMs [91].

3.2.3 Queueing model

Every frame n in the CWN, $\mu_i(n) + \mu_{is}(n)$ application requests are removed from the buffer of terminal i. Let $q_i(n)$ denote the queue length at terminal i and $\vec{Q}(n) = [q_1(n), q_2(n), ..., q_{|I|}(n)]$ represent the vector of queue lengths at the terminals in frame n. The queuing dynamics in the CWN are then given as

$$q_i(n+1) = q_i(n) + a_i(n) - [\mu_i(n) + \mu_{is}(n)]. \tag{26}$$

In the cognitive wireless networks, the corresponding queuing process is given as

$$q_i(n+1) = q_i(n) + a_i(n) - I(n)[\mu_i(n) + I_r(n)\mu_{is}(n)] + [1 - I(n)]\mu_{i0}(n). \tag{27}$$

In addition, let $y_i(n) = q_i(n) + a_i(n)$ and $\vec{Y}(n)$ is the vector of $y_i(n)$s.

At each server s, the delegated requests can be stored into a buffer reserved for terminal i at server s before the requests are processed at the server. We use $\hat{q}_s^i(n)$ to denote the queue length of terminal i at server s, $\widehat{\mathbf{Q}}(n) = [\hat{q}_1^1(n), \hat{q}_2^1(n), ..., \hat{q}_{|\mathscr{I}|}^1(n)\ ; ...;$ $\hat{q}_1^{|\mathscr{I}|}(n), \hat{q}_2^{|\mathscr{I}|}(n), ...\ \hat{q}_{|\mathscr{I}|}^{|\mathscr{I}|}(n)]$ denotes the $|\mathscr{I}| \times |\mathscr{S}|$ matrix of the queue lengths at each server s and $\widehat{Q}_i(n) = [\hat{q}_1^i(n), \hat{q}_2^i(n), ..., \hat{q}_{|\mathscr{I}|}^i(n)]$ represents the ith row of $\widehat{\mathbf{Q}}(n)$. Let $\hat{\mu}_s^i(n)$ represent the service rate [requests/frame] server s provides to terminal i in frame n. The queueing dynamics for the application requests of terminal i at server s for both PSP and SSP cognitive networks is given as

$$\hat{q}_s^i(n+1) = \hat{q}_s^i(n) + I(n) I_{\mathrm{r}}(n) \mu_{is}(n) - \hat{\mu}_s^i(n). \tag{28}$$

For the CWN, $\hat{q}_s^i(n+1)$ is written as

$$\hat{q}_s^i(n+1) = \hat{q}_s^i(n) + \mu_{is}(n) - \hat{\mu}_s^i(n). \tag{29}$$

Finally, let $\hat{\mu}_s(n) = \sum_{i \in \mathscr{I}} \hat{\mu}_s^i(n)$ represent the total service rate at server s, and $\hat{q}_s(n) = \sum_{i \in \mathscr{I}} \hat{q}_s^i(n)$ denote the sum of queue lengths at server s.

3.3 Unified problem formulation

In order to derive a unified optimization problem for both CWN and PSP/SSP cognitive wireless networks, one should note that the service rates for the PSP/SSP cognitive networks can be derived from the service rates of the CWN. When the number of requests transmitted from terminal i to server s and the number of requests processes at terminal i in the CWN are given by $\mu_{is}(n)$ and $\mu_i(n)$, respectively, the corresponding service rates for PSP and SSP cognitive networks are defined as

$$\mu_{is}(n)^* = \mu_{is}(n) p[I(n) = 1] p[I_{\mathrm{r}}(n) = 1] \tag{30}$$

$$\acute{\mu}_i(n)^* = \mu_i(n) p[I(n) = 1] + \mu_{i0}(n) p[I(n) = 0] = \mu_i(n)^* + \mu_{i0}(n)^*, \tag{31}$$

where $\mu_i(n)^* = \mu_i(n) p[I(n) = 1]$ and $\mu_{i0}(n)^* = \mu_{i0}(n) p[I(n) = 0]$.

Given (30) and (31), the unified power consumption and queueing dynamics for both PSP and SSP cognitive networks as well as for CWN are

$$P_i(n) = \alpha_i \acute{\mu}_i(n)^*, \tag{32}$$

$$P_{is}(n) = \frac{\mu_{is}(n)p[I(n)=1]\alpha_{is}}{|h_{is}(n)|^2}, \tag{33}$$

$$q_i(n+1) = q_i(n) + a_i(n) - [\acute{\mu}_i(n)^* + \mu_{is}(n)^*] \tag{34}$$

for each terminal i and

$$\hat{q}_s^i(n+1) = \hat{q}_s^i(n) + \mu_{is}(n)^* - \hat{\mu}_s^i(n) \tag{35}$$

for each terminal i at server s.

A specific control action at terminal i is a decision on how many applications are processed at the terminal, how many requests are forwarded to server s, and which specific server s is hosting the terminal i. Let $\mathscr{U}(n)$ denote the set of control actions available at the terminals in frame n, and $U_i(n) = \{\acute{\mu}_i(n)^*, \mu_{is}(n)^*, b_{is}(n)\} \in \mathscr{U}(n)$ represents a specific control action at terminal i in frame n. In addition, $\vec{U}(n) = [U_1(n), U_2(n), ..., U_{|\mathscr{I}|}(n)]$ is used to represent the vector of control actions in frame n.

The control action at each server s includes selecting the CPU frequency, that affects the power consumption $\hat{P}_s(n)$, as well as CPU resource distribution among different VMs that host the terminals running on that server. This allocation is subject to the available control options at each server s. For example, the controller may allocate different fractions of CPU to the VMs in that frame. We use $\hat{\mathscr{U}}(n)$ to denote the set of all control actions available at server s. Let $\hat{U}_s(n) = \{\hat{\mu}_s(n)\} \in \hat{\mathscr{U}}(n)$ denote a particular control action taken at server s in frame n under any policy and $\hat{P}_s(n)$ is the corresponding power consumption. The vector of control actions at the data center is given as $\hat{U}(n) = [\hat{U}_1(n), \hat{U}_2(n), ..., \hat{U}_{|\mathscr{S}|}(n)]$.

Let $X(n) = \{\vec{Y}(n), \widehat{\mathbf{Q}}(n), \mathbf{H}(n)\}$ represent the state of the system in frame n with countable state space $\mathscr{X}$, where $\mathbf{H}(n) = [|h_{11}(n)|^2, |h_{12}(n)|^2, ..., |h_{1|\mathscr{S}|}(n)|^2; |h_{21}(n)|^2, |h_{22}(n)|^2, ..., |h_{2|\mathscr{S}|}(n)|^2; ...; |h_{|\mathscr{I}|1}(n)|^2, |h_{|\mathscr{I}|2}(n)|^2, ..., |h_{|\mathscr{I}||\mathscr{S}|}(n)|^2]$ denote $|\mathscr{I}| \times |\mathscr{S}|$ channel gain matrix in frame n. We use $D_X(n) = \{\vec{U}(n), \hat{U}(n)\}$ to denote the control input, i.e., the action, in fame n, when the state of the system is $X(n)$. At the beginning of each frame n, the network controller decides upon the value of $D_X(n)$ depending on the current state of the system $X(n)$. The control input $D_X(n)$ takes values in a general state space $\mathscr{D}_X(n)$, which represents all the feasible control options in state $X(n)$. Starting from state X, let $\pi = \{D_X(1), D_X(2), ...\}$ denote the policy, i.e., the sequence of actions. We use Π to denote the space of all such policies and $\pi \in \Pi$.

It is important to note that the availability of the software resources could be added here to the system model by simply introducing a binary variable

$$\varphi_i(n) = \begin{cases} 1; & \text{If terminal } i \text{ has the necessary} \\ & \text{software to process the applications.} \\ 0; & \text{Otherwise.} \end{cases}$$

and rewriting the state as $X(n) = \{\vec{Y}(n), \widehat{\mathbf{Q}}(n), \mathbf{H}(n), \vec{\varphi}(n)\}$, where $\vec{\varphi}(n) = [\varphi_1(n), ..., \varphi_{|\mathscr{I}|}(n)]$ is the vector of variables $\varphi_i(n)$. If $\varphi_i(n) = 0$, application requests cannot be processed at terminal i in frame n.

Let δ_i represent a non-negative weight used as a normalizing parameter. The goal is to map from the current $X(n)$ to an sequence of $D_X(n)$, that solves the following optimization problem:

$$\begin{aligned} \underset{\pi\in\Pi}{\text{maximize}} \quad & \lim_{n\to\infty} \frac{1}{n} \sum_{\eta=0}^{n-1} \sum_{i\in\mathscr{I}} \mathbb{E}_X^{\pi} \Big\{ \acute{\mu}_i(\eta)^* + \sum_{s\in\mathscr{S}} b_{is}(\eta)\mu_{is}(\eta)^* - \\ & \delta_i \frac{P_i^{\text{tot}}(\eta)}{P^{\max}} \Big\} - \lim_{n\to\infty} \frac{1}{n} \sum_{\eta=0}^{n-1} \sum_{s\in\mathscr{S}} \mathbb{E}_X^{\pi} \{ \hat{P}_s(\eta) \} \end{aligned} \tag{36}$$

subject to

$$\lambda_i \in \Lambda_{\text{T}},$$

$$q_i(\eta) < \infty \quad \text{and} \quad \hat{q}_s^i(\eta) < \infty,$$

$$P_i^{\text{tot}}(\eta) \leq P^{\max} \quad \text{and} \quad \hat{P}_s(\eta) \leq \hat{P}^{\max}.$$

The constraints are valid for all $i \in \mathscr{I}$ and $s \in \mathscr{S}$ and Λ_{T} represents network stability region presented later in Section 3.5.

The objective in (36) is a constrained dynamic optimization problem and it maximizes the joint utility of the sum throughput of the applications processed at the terminals and minimizes the overall power usage both at the terminals and at the data center. It allows the design of resource allocation policies that adjust to workload and channel variations. For example, if the current workload is small, then this objective encourages scaling down the instantaneous capacity in the servers in order to achieve energy savings. Similarly if the current workload is large, the objective encourages scaling up the instantaneous capacity by higher power consumption. In addition, (36) encourages to delay some parts of input traffic by scheduling more packets in good channel states, and less in poor conditions in order to achieve the maximum long-term throughput with minimum power consumption.

3.4 Unified control policy

In this section, we propose a dynamic control policy that solves the constrained dynamic optimization problem in (36). Every frame n, the policy uses the current QSI and CSI to define resource allocation decisions $U_i(n)$ and $\hat{U}_s(n)$ for each terminal i and server s. As the exact dynamic programming is computationally very complex, we propose a solution, where control actions can be calculated separately for each terminal i and server s.

3.4.1 Resource allocation at the terminals

Let $X_i(n) = \{y_i(n), \widehat{Q}_i(n), \vec{H}_i(n)\}$ represent the state of terminal i in frame n with countable state space $\mathscr{X}_i$. In addition, we use $U_{X_i}(n) = \{\acute{\mu}_i(n)^*, \mu_{is}(n)^*, \vec{B}_i(n)\}$ to denote the control input, i.e., action, at terminal i in frame n in state $X_i(n)$. The control input $U_{X_i}(n)$ takes values in a general state space $\mathscr{U}_{X_i}(n)$, which represents all the feasible resource allocation options available in state $X_i(n)$ in frame n. By feasible options we mean the set of control actions that satisfy the power and the queue constraints, as we cannot transmit more application requests than there are in the queue. Let $\pi_i = \{U_{X_i}(0), U_{X_i}(1), ...\}$ denote the policy, i.e., the sequence of actions, at terminal i, and Π_i represent the space of all such policies.

For each terminal i, the goal is to map from the current QSI and CSI to a policy $\pi_i^\star \in \Pi_i$ that stabilizes the system and solves the following control problem:

$$\begin{aligned} \underset{\pi_i \in \Pi_i}{\text{maximize}} \quad & \lim_{n\to\infty} \frac{1}{n} \sum_{\eta=0}^{n-1} \mathbb{E}_{X_i}^{\pi_i}\{T_i(\eta) + S_i(\eta)\} \\ \text{subject to} \quad & \lim_{n\to\infty} \frac{1}{n} \sum_{\eta=0}^{n-1} \mathbb{E}_{X_i}^{\pi_i}\left\{\frac{P_i^{\text{tot}}(\eta)}{P^{\max}}\right\} \leq 1. \end{aligned} \tag{37}$$

In (37),

$$T_i(\eta) = [y_i(\eta) - \sum_{s\in\mathscr{S}} b_{is}(\eta)\hat{q}_s^i(\eta)] \frac{\sum_{s\in\mathscr{S}} b_{is}(\eta)\mu_{is}(\eta)^*}{\mu_{is}^{\max}}, \tag{38}$$

$$S_i(\eta) = y_i(\eta)[\acute{\mu}_i(\eta)^* + \sum_{s\in\mathscr{S}} b_{is}(\eta)\mu_{is}(\eta)^*] \tag{39}$$

and the maximum number of application requests that can be delivered from terminal i to server s in one frame is

$$\mu_{is}^{\max} = \max_{\{s\in\mathscr{S}, \vec{H}_i \in \mathscr{H}_i\}} \frac{P^{\max}|h_{is}|^2}{\alpha_{is}}. \tag{40}$$

One should note that based on the definition of H_i^{e} for PSP/SSP cognitive networks in Subsection 3.2.1, $\mu_{is}^{\max}$ gets the same value for both the PSP/SSP cognitive network and the CWN. Equation (37) maximizes the long-term average throughput of the terminals while keeping the energy cost and queues low. For example, high power computationally intensive application requests at the terminal can be delegated to the hosting server in order to achieve energy savings at the terminal. If the backlog value at the terminal i is larger than the backlog of terminal i at server s, the objective in (37) encourages the terminal to delegate its requests to be processed at the servers.

Formulation as a Markov decision process

We first convert the constrained dynamic optimization problem in (37) into an unconstrained problem (UP) and then find the control policy for this UP [15], [41], [42], [92].

The set of feasible actions U_{X_i} in each state $X_i = \{y_i, \widehat{Q}_i, \vec{H}_i\}$ is the set of all actions $U_{X_i} = \{\acute{\mu}_i^*, \mu_{is}^*, \vec{B}_i\}$ that satisfy the power and the queue constraints as we cannot transmit more packets than there are in the queue, i.e., $\acute{\mu}_i^* + \mu_{is}^* \leq y_i$ and $P_i^{\mathrm{tot}} \leq P^{\max}$. After taking an action $U_{X_i} = \{\acute{\mu}_i^*, \mu_{is}^*, \vec{B}_i\}$, the following state is given as $Z_i = \{q_i, \widehat{Y}_i, \vec{H}_i\}$, where $\widehat{Y}_i = [\hat{y}_1^i, ..., \hat{y}_{|\mathscr{S}|}^i]$ and $\hat{y}_s^i = \hat{q}_s^i + b_{is}\mu_{is}^*$. Based on (26) and (29), we get this by noting that $y_i - (\acute{\mu}_i^* + \mu_{is}^*) = q_i$ and $\hat{q}_s^i + \mu_{is}^* = \hat{y}_s^i$. It is important to note that for each state $X_i = \{y_i, \widehat{Q}_i, \vec{H}_i\}$ with equal $\widehat{Q}_i$ and $\vec{H}_i$, where $q_i \in \{0, 1, ..., y_i\}$, $a_i \in \{0, 1, ..., y_i\}$ and $q_i + a_i = y_i$, the set of feasible actions and following states are the same. Thus, state $Z_i = \{q_i, \widehat{Y}_i, \vec{H}_i\}$ is *equivalent* to a state $X_i = \{y_i, \widehat{Q}_i, \vec{H}_i\}$, if the channels are the same and both q_i and a_i take values within the set $\{0, 1, ..., y_i\}$ so that $q_i + a_i = y_i$ and $\hat{q}_s^i$ takes values within the set $\{0, 1, ..., \hat{y}_s^i\}$ so that $\hat{y}_s^i = \hat{q}_s^i + b_{is}\mu_{is}^*$ for each server s. When $a_i = 0$ and $b_{is}\mu_{is}^* = 0$ for all $s \in \mathscr{S}$, we have $y_i = q_i$ and $\widehat{Q}_i = \widehat{Y}_i$. Then, $X_i = \{y_i, \widehat{Q}_i, \vec{H}_i\} = \{q_i, \widehat{Y}_i, \vec{H}_i\} = Z_i$. For example, let us consider a system with a terminal and 2 servers. In state $X_i = \{y_i, \widehat{Q}_i, \vec{H}_i\}$, we let $y_i = 3$ and $\widehat{Q}_i = [\hat{q}_1^i, \hat{q}_2^i] = [1, 2]$. Then, $q_i = \{0, 1, ..., 3\}$, $a_i = \{0, 1, ..., 3\}$, $q_i + a_i = 3$ and $[\hat{y}_1^i, \hat{y}_2^i] = [1, 2]$. When $a_i = 0$ and $b_{is}\mu_{is}^* = 0$, $y_i = q_i = 3$ and $\hat{q}_s^i = \hat{y}_s^i$. Now we have $X_i = Z_i$. This property is important when calculating the value functions in (48), as $W^l(X_i) = W^l(Z_i)$, if X_i is *equivalent* to Z_i. Let $p(Z_i|X_i, U_{X_i})$ denote the transition probability from state X_i to state Z_i with action U_{X_i}.

For a policy π_i, define the reward D_i and cost functions E_i as

$$D_i = \lim_{n\to\infty} \frac{1}{n} \sum_{\eta=0}^{n-1} \mathbb{E}_{X_i}^{\pi_i} \{T_i(\eta) + S_i(\eta)]\} \tag{41}$$

and

$$E_i = \lim_{n\to\infty} \frac{1}{n} \sum_{\eta=0}^{n-1} \mathbb{E}_{X_i}^{\pi_i} \Big\{ \frac{P_i^{\text{tot}}(\eta)}{P^{\max}(\eta)} \Big\}. \tag{42}$$

Let Π_i^E denote the set of all admissible control policies $\pi_i \in \Pi_i$, which satisfy the constraint $E_i(\eta) \leq 1$ in every frame η. Then, (37) can be restated as a constrained optimization problem given as

$$\text{maximize} \;\; D_i; \;\; \text{subject to} \;\; \pi_i \in \Pi_i^E. \tag{43}$$

The problem (43) can be converted into a family of unconstrained optimization problems through a Lagrangian relaxation [93]. The corresponding Lagrangian function for any policy $\pi_i \in \Pi_i$ and for every $\beta_i \geq 0$ can now be defined as

$$J_\beta^{\pi_i}(X_i) = \lim_{n\to\infty} \frac{1}{n} \sum_{\eta=0}^{n-1} \mathbb{E}_{X_i}^{\pi_i} \big\{ T_i(\eta) + S_i(\eta) - \beta_i E_i(\eta) \big\}. \tag{44}$$

Given $\beta_i \geq 0$, the unconstrained optimization problem is defined as

$$\text{maximize} \;\; J_\beta^{\pi_i}(X_i) \;\; \text{subject to} \;\; \pi_i \in \Pi_i. \tag{45}$$

A policy for unconstrained problem is also valid for the original constrained control problem, when β_i is appropriately chosen [15], [93].

The problem given in (45) is a standard MDP with the maximum average reward criterion. For each initial state $X_i \in \mathscr{X}_i$, define a corresponding discounted reward MDP with value function

$$W_\alpha(X_i) = \underset{\pi_i \in \Pi_i}{\text{maximize}} \;\; \sum_{n=0}^{\infty} \mathbb{E}_{X_i}^{\pi_i} \big\{ \alpha^n R[U_{X_i}(n), X_i(n)] \big\} \tag{46}$$

where the discount factor $\alpha \in (0,1)$, and a reward from taking an action $U_{X_i}(\eta)$ in state $X_i(\eta)$ is defined as

$$R[U_{X_i}(n), X_i(n)] = T_i(n) + S_i(n) - \beta_i E_i(n). \tag{47}$$

$W_\alpha(X_i)$ is defined as the total expected discounted utility for discount factor α [94]. One way to solve (46) is to use value iteration algorithm (VIA) [15], [94], [95].

For notational simplicity, we suppress the subscript α. The solution to (46), i.e., the value functions $W^\star(X_i)$ for each initial state X_i and the corresponding control sequences $\pi_i^\star \in \Pi_i$ can be solved with the following value iteration algorithm [15], [95]:

$$W^{l+1}(X_i) = \max_{U_{X_i} \in \mathscr{U}_{X_i}} \{ R(U_{X_i}, X_i) + \alpha \sum_{Z_i \in \mathscr{Z}_i} p(Z_i | X_i, U_{X_i}) W^l(Z_i) \}. \tag{48}$$

In (48), $\mathscr{Z}_i \subset \mathscr{X}_i$ is the set of feasible states that follow state X_i by taking an action U_{X_i}, and l denotes the iteration index. For each initial state X_i, define the control action in each state X_i as

$$\arg \max_{U_{X_i} \in \mathscr{U}_{X_i}} \left\{ R(U_{X_i}, X_i) + \alpha \sum_{Z_i \in \mathscr{Z}_i} p(Z_i | X_i, U_{X_i}) W^\star(Z_i) \right\}. \tag{49}$$

3.4.2 Resource allocation at the servers

Let $\hat{X}_s(n) = [\hat{y}_s^1(n), ..., \hat{y}_s^{|I|}(n)]$ represent the system state at server s in frame n with countable state space $\hat{\mathscr{X}}_s$. Let $\hat{U}_{\hat{X}_s}(n) = \{[\hat{\mu}_s^1(n), ..., \hat{\mu}_s^{|\mathscr{I}|}(n)]\}$ denote the particular control action in state $\hat{X}_s(n)$, and $\hat{\mathscr{U}}_{\hat{X}_s}(n)$ is the set of feasible resource allocation options in each state $\hat{X}_s(n)$. In addition, we use $\hat{\pi}_s = \{\hat{U}_{\hat{X}_s}(1), \hat{U}_{\hat{X}_s}(2), ...\}$ to denote the sequence of control actions at server s and $\hat{\Pi}_s$ represents the set of all such policies.

For each terminal s, map from the current queue and channel states to a sequence of actions that stabilizes the system and solves the following optimization problem:

$$\begin{aligned} &\underset{\hat{\pi}_s \in \hat{\Pi}_s}{\text{maximize}} \quad \lim_{n\to\infty} \frac{1}{n} \sum_{\eta=0}^{n-1} \sum_{i \in \mathscr{I}} \mathbb{E}_{\hat{X}_s}^{\hat{\pi}_s} \{\hat{y}_s^i(\eta) \hat{\mu}_s^i(\eta)\} \\ &\text{subject to} \quad \widehat{P}_{\min} \leq \lim_{n\to\infty} \frac{1}{n} \sum_{\eta=0}^{n-1} \mathbb{E}_{\hat{X}_s}^{\hat{\pi}_s} \{\widehat{P}_s(\eta)\} \leq \widehat{P}_{\max}. \end{aligned} \tag{50}$$

The objective encourages allocating bigger fractions of CPU to the VMs of the terminals with the biggest backlog values at the server. If the current backlog value of terminal i at server s is inside the instantaneous capacity region, then this objective also encourages allocating less CPU to the VMs of the terminals with low backlog values and/or run CPU at slower speeds to achieve energy savings at the server.

Formulation as a Markov decision process

The set of feasible actions in each state $\hat{X}_s = [\hat{y}_s^1, ..., \hat{y}_s^{|I|}]$ is the set of all $\{[\hat{\mu}_s^1, ..., \hat{\mu}_s^{|\mathscr{I}|}]\}$ that satisfy $\hat{\mu}_s^i \leq \hat{y}_s^i$ and $\hat{P}_s \leq \hat{P}^{\max}$. After taking an action $\hat{U}_{\hat{X}_s}$, the following state is given as $\hat{Z}_s = \{[\hat{q}_s^1, ..., \hat{q}_s^{|I|}]\}$. State $\hat{Z}_s$ that is equivalent to a state $\hat{X}_s$, where $\hat{q}_s^i \in \{0, 1, ..., \hat{y}_s^i\}$, $b_{is}\mu_{is}^* \in \{0, 1, ..., \hat{y}_s^i\}$ and $\hat{q}_s^i + b_{is}\mu_{is}^* = \hat{y}_s^i$, as described in Subsubsection 3.4.1. Let $p(\hat{Z}_s | \hat{X}_s, \hat{U}_{\hat{X}_s})$ denote the transmission probability from state $\hat{X}_s$ to state $\hat{Z}_s$ with action $\hat{U}_{\hat{X}_s}$. Just as in Subsubsection 3.4.1, (50) can now be solved by converting it into a MDP and by finding the control policy for this MDP using the VIA.

3.5 Achievable rates

In this Section, we characterize the fundamental throughput limitations and present the unified capacity/stability region of the system given in Fig. 1 for both SSP and PSP cognitive networks as well as for the CWN. As the optimization can be solved separately for each terminal i and server s, the supportable arrival rate regions can also be derived separately for the two cases.

3.5.1 Unified arrival rate region at the terminals

Let g_i denote the long-term average number of application requests that can be supported at each terminal i in the CWN. We use c_i to denote the long-term average number of application requests processed at terminal i, c_{is} represents the long-term average number of application requests delivered from terminal i to server s and $g_i = c_i + \sum_{s\in\mathscr{S}} c_{is}$.

Given c_i and c_{is} for the CWN, the long-term average number of application requests processed at terminal i and the long-term average number of application requests delivered from terminal i to server s for the cognitive wireless networks are respectively given as

$$c_{is}^* = c_{is}\, p(I=1)\, p(I_{\mathrm{r}}=1) \tag{51}$$

$$\acute{c}_i^* = c_i\, p(I=1) + c_{i0}\, p(I=0) = c_i^* + c_{i0}^*, \tag{52}$$

where $c_i^* = c_i p(I=1)$ and $c_{i0}^* = c_{i0} p(I=0)$. Here c_{i0}^* represents the long-term average number of requests processed at terminal i, when there is no channel available between terminal i and server s, i.e., $\vec{H}_i^e = 0$. Let $g_i^* = \acute{c}_i^* + \sum_{s\in\mathscr{S}} c_{is}^*$ denote the long-term average number of application requests that can be supported at terminal i in PSP/SSP cognitive networks.

Due to the time varying channel conditions between terminal i and the servers, g_i^* must be averaged over all possible channel states. Moreover, for the given channel states, g_i^* is not fixed and depends on control policy $\pi_i \in \Pi_i$ for choosing the control actions. Thus, numerical calculation of all supportable rates g_i^* is computationally very challenging.

However, based on (21) and (22), the supportable arrival rate region at the terminals can also be defined by considering only the set of policies, where each terminal transmits at full power in each frame n. Let $\mathscr{O}_{H_i} \subset \mathscr{U}_{X_i}$ represent the set of possible options to allocate the total power $P^{\max}$ at each terminal i in channel state $\vec{H}_i$. In addition, we use $O_{H_i} \in \mathscr{O}_{H_i}$ to denote a total power allocation action at terminal i, when the system is in

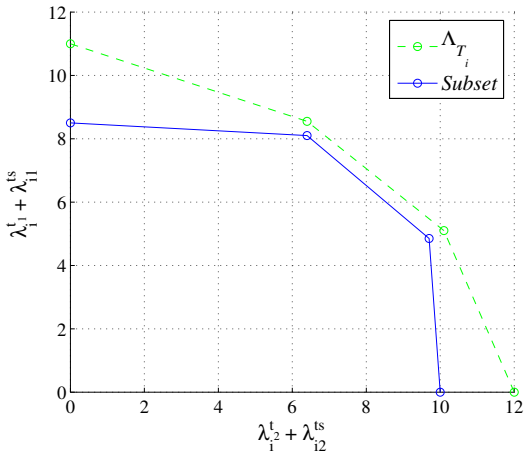

Fig. 2. The rate region $\lambda_i^{t_2} + \lambda_{i2}^{ts}$ vs. $\lambda_i^{t_1} + \lambda_{i1}^{ts}$ and the subregion λ_{i2}^{ts} vs. λ_{i1}^{ts}.

channel state $\vec{H}_i$. The long-term average transmission rate of terminal i for the full power policies is given by $g^*_{\max_i}$. The set of all full power long-term average transmission rates $g^*_{\max_i}$ that a terminal can be configured to support is now given as

$$\Gamma^* = \sum_{\vec{H}_i \in \mathscr{H}_i^{\prime}} \pi_{H_i} \mathrm{Conv}\{\mu_i(O_{H_i}, \vec{H}_i)^* + \sum_{s \in \mathscr{S}} b_{is}\mu_{is}(O_{H_i}, \vec{H}_i)^* | O_{H_i} \in \mathscr{O}_{H_i}\} + p(I=0)\mu_{i0}^{\max}, \tag{53}$$

where

$$\mu_{i0}^{\max} = P^{\max}/\alpha_i \tag{54}$$

is the maximum number of requests that can be processed at terminal i, when there is no channel available between terminal i and server s. For the PSP and SSP cognitive networks, $p(I=0) = p_0^{\mathrm{P}}$ and $p(I=0) = p_0^{\mathrm{S}}$, respectively. In the CWN, $p(I=0) = 0$. In (53), addition and scalar multiplication of sets are used, and $\mathrm{Conv}\{\mathscr{B}\}$ represents the convex hull of the set $\mathscr{B}$ that is defined as the set of all convex combinations $p_1 v_1 + p_2 v_2 + ... + p_j v_j$ of elements $v_j \in \mathscr{V}$, where p_js are probabilities summing to 1.

The throughput region Γ^* can be viewed as the set of all long-term full power average service rates $g^*_{\max_i}$ that the terminal can be configured to support. Thus, the unified supportable rate region Λ_{T} at the terminals for both the PSP and SSP cognitive

networks as well as for the CWN is the set of all average arrival rates vectors $\vec{\lambda} = [\lambda_1, \lambda_2, ..., \lambda_{|\mathscr{I}|}]$ for which there exists a control policy π_i that satisfies

$$\lambda_i \leq \lim_{n\to\infty} \frac{1}{n} \sum_{\eta=1}^{n-1} \mathbb{E}_{X_i}^{\pi_i}\{\mu_i(\eta)^* + \sum_{s\in\mathscr{S}} b_{is}(\eta)\mu_{is}(\eta)^*\} + p(I=0)\mu_{i0}^{\max} \leq g_{\max_i}^* \qquad (55)$$

for some $g_{\max_i}^* \in \Gamma^*$, as rates below each point in Γ^* can likewise be supported. Specifically, $\vec{\lambda}$ is in the region Λ_{T} if there exists a average service rate vector such that there exists a control process which supports the rates $\vec{\lambda}$.

For the CWN, we write λ_i as $\lambda_i = \lambda_i^{\mathrm{t}} + \sum_{s\in\mathscr{S}} \lambda_{is}^{\mathrm{ts}}$, where λ_i^{t} denotes the average number of supported input requests at terminal i that are processed at terminal i, and $\lambda_{is}^{\mathrm{ts}}$ represents the average number of supported input requests at terminal i that are forwarded from terminal i to server s. In addition, let $\lambda_i^{\mathrm{t}_s}$ denote the average number of supportable input requests processed at terminal i, when $b_{is} = 1$, and $\lambda_i^{\mathrm{t}} = \sum_{s\in\mathscr{S}} \lambda_i^{\mathrm{t}_s}$. In order to avoid multidimensional illustration of the results, we fix $|\mathscr{I}| = |\mathscr{S}| = 2$. For the channel model given in Section 3.8, the supportable rate region $\lambda_i^{\mathrm{t}_2} + \lambda_{i2}^{\mathrm{ts}}$ vs. $\lambda_i^{\mathrm{t}_1} + \lambda_{i1}^{\mathrm{ts}}$ is plotted as a dashed line in Fig. 2 and denoted as Λ_{T_i}. For comparison, the subset of the region Λ_{T_i} in Fig. 2, illustrates the supportable arrival rate region for the channels between terminal i and servers, i.e., $\lambda_i^{\mathrm{t}_1} = \lambda_i^{\mathrm{t}_2} = 0$.

Let $\lambda_i^{\max}$ denote the maximum average number of requests that can be supported at terminal i in the CWN. It can be seen in Fig. 2, that $\lambda_i^{\max} = 8 + 7 = 15$. We have $\lambda_i^{\max} = \lambda_{\max_i}^{\mathrm{t}} + \sum_{s\in\mathscr{S}} \lambda_{\max_{is}}^{\mathrm{ts}}$, where $\lambda_{\max_i}^{\mathrm{t}}$ denote the maximum number of supported input requests at the terminal i processed at terminal i and $\lambda_{\max_{is}}^{\mathrm{ts}}$ represents the maximum number of supported input requests at terminal i forwarded from terminal i to server s. In Fig. 2, it can be seen that $\lambda_{\max_i}^{\mathrm{t}} = 0.5$ and $\sum_{s\in\mathscr{S}} \lambda_{\max_{is}}^{\mathrm{ts}} = 8 + 6.5 = 14.5$. Given $\lambda_i^{\max}$, the maximum supportable arrival rate at terminal i for the PSP and SSP cognitive networks is given as

$$\lambda_{\max_i}^{\mathrm{cn}} = \lambda_{\max_i}^{\mathrm{t}} p(I=1) + \sum_{s\in\mathscr{S}} \lambda_{\max_{is}}^{\mathrm{ts}} p(I=1)p(I_{\mathrm{r}}=1) + p(I=0)\mu_{i0}^{\max}. \qquad (56)$$

For the channel model of the CWN given in Section 3.8, the unified supportable arrival rate region at terminals (Λ_{T}) for both the PSP and SSP cognitive networks as well as for the CWN is now illustrated in Fig. 3.

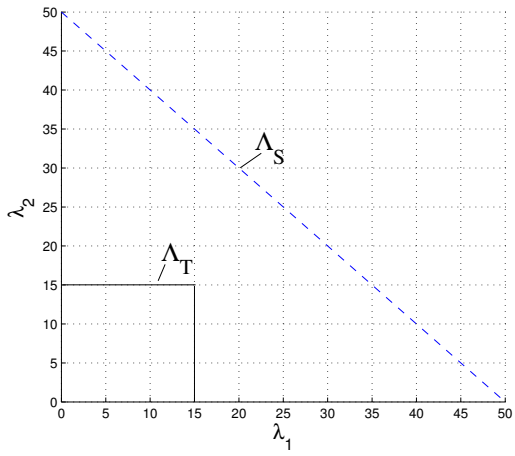

Fig. 3. The unified supportable arrival rate region at the terminals ($\Lambda_{\mathbf{T}}$) and at the server ($\Lambda_{\mathbf{S}}$).

3.5.2 Unified arrival rate region at servers

Let $\hat{g}_s^i$ denote the long-term average number of application requests of terminal i processed at server s, and $\hat{g}_s = \sum_{i\in\mathscr{I}} \hat{g}_s^i$ is the long-term average supportable rate at server s. The long-term average number of application requests $\hat{g}_s$ is not fixed and depends on control policy for choosing the actions.

Let Λ_{S} represent the supportable arrival rate region at server s. In order to calculate Λ_{S}, we consider only the set of policies that consume the whole $\hat{P}^{\max}$ at server s in each frame n. We use $\hat{\mathscr{O}}_s$ to represent the set of possible full power allocation options at server s, and $\hat{O}_s \in \hat{\mathscr{O}}_s$ denotes a full power allocation action at server s. One should note that $\hat{\mathscr{O}}_s \subset \hat{\mathscr{U}}_s$. Let $\hat{g}_s^{\max}$ denote the long-term full power average number of requests processed at server s. The set of full power average number of requests that can be supported at server s is

$$\hat{\Gamma} = \mathrm{Conv}\{\hat{\mu}_s^1(\hat{O}_s) + \hat{\mu}_s^2(\hat{O}_s) + ..., + \hat{\mu}_s^{|\mathscr{I}|}(\hat{O}_s) | \hat{O}_s \in \hat{\mathscr{O}}_s\}. \tag{57}$$

Specifically, the throughput region $\hat{\Gamma}$ can be viewed as the set of all full power long-term average service rates $\hat{g}_s^{\max}$ that a server can be configured to support. Thus, the supportable arrival rate region Λ_{S} at server s is the set of all average arrival rates $\sum_{i\in\mathscr{I}} \lambda_{is}^{\mathrm{ts}}$

for which there exists a control policy $\hat{\pi}_s$ that satisfies

$$\sum_{i\in\mathscr{I}} \lambda_{is}^{\mathrm{ts}} \leq \lim_{n\to\infty} \frac{1}{n} \sum_{\eta=1}^{n-1} \mathbb{E}\{\hat{\mu}_s(\eta)\} \leq \hat{g}_s \leq \hat{g}_s^{\max} \tag{58}$$

for some $\hat{g}_s^{\max} \in \hat{\Gamma}$ as rates below each point in $\hat{\Gamma}$ can likewise be supported.

For comparison, the supportable arrival rate region at server s, Λ_{S}, is illustrated in Fig. 3 together with Λ_{T}. Since Λ_{T} is a subset of Λ_{S}, it is clear that server s can support all arrival rates λ inside Λ_{T}. Thus, the network stability region Λ is equal to Λ_{T}. Stability region is unique for each network and it should not be mixed up with the stability region of a specific resource allocation policy. The stability region of a resource allocation policy is a closure of the set of arrival rate vectors $\vec{\lambda}$ that the policy can stably support, and it is a subset of the network capacity region [12].

3.6 Complexity analysis

In this section, we analyse the complexity of the dynamic control policy proposed in Section 3.4.

In order to calculate the control policy in (49), we first need to calculate the rewards in (47) and then the value functions in (48). It is easy to see that the complexity of calculating the control policy depends not only on the sizes of $\mathscr{X}_i$ and $\hat{\mathscr{X}}_s$ but also on the number of feasible control options in each state $X_i \in \mathscr{X}_i$ and $\hat{X}_s \in \hat{\mathscr{X}}_s$. We start with defining the cardinality of $\mathscr{X}_i$ and $\hat{\mathscr{X}}_s$.

Let $|\mathscr{X}_i|$ and $|\hat{\mathscr{X}}_s|$ denote the number of states in $\mathscr{X}_i$ and $\hat{\mathscr{X}}_s$, respectively. In addition, let $|\mathscr{H}_i|$ denote the number of channel states in state space $\mathscr{H}_i$. For arrival rates inside Λ_{T}, we have $\limsup_{n\to\infty} y_i(\eta) = y_i^{\max}$ and $\limsup_{n\to\infty} \hat{y}_s^i(\eta) = \hat{y}_s^{\max}$ for all $i \in \mathscr{I}$ and $s \in \mathscr{S}$. The total number of states at terminal i is

$$|\mathscr{X}_i| = (y_i^{\max}+1)|\mathscr{H}_i|(\hat{y}_s^{\max}+1)^{|\mathscr{S}|} \tag{59}$$

and the total number of states at servers s

$$|\hat{\mathscr{X}}_s| = (\hat{y}_s^{\max}+1)^{|\mathscr{I}|}. \tag{60}$$

The rewards in (47) need to be calculated for each action $U_{X_i} \in \mathscr{U}_{X_i}$ in each state $X_i \in \mathscr{X}_i$. Let $|\mathscr{U}_{X_i}|$ and $|\hat{\mathscr{U}}_{\hat{X}_s}|$ denote the number of feasible control actions in each state $X_i \in \mathscr{X}_i$ and $\hat{X}_s \in \hat{\mathscr{X}}_s$, respectively. In addition, we use $\mu_{X_i}^{\max}$ to represent the maximum

number of application requests that can be removed from the buffer of terminal i with power $P^{\max}$ in state X_i. The number of feasible actions in state X_i is then given as

$$|\mathscr{U}_{X_i}| = (|\mathscr{S}|+1)\min\{y_i, \mu_{X_i}^{\max}\}+1, \tag{61}$$

and the number of feasible actions in state $\hat{X}_s$ as

$$|\hat{\mathscr{U}}_{\hat{X}_s}| = |\mathscr{I}|\min\{\sum_i \hat{y}_s^i, \hat{\mu}_s^{\max}\}+1, \tag{62}$$

where $\hat{\mu}_s^{\max} = \hat{P}^{\max}/\alpha_s$. The total number of calculated rewards at terminal i and server s are now given as $\sum_{|\mathscr{X}_i|}|\mathscr{U}_{X_i}|$ and $|\hat{\mathscr{X}_s}||\hat{\mathscr{U}}_{\hat{X}_s}|$, respectively.

After calculating all the rewards, we get the value functions $W^\star$ by calculating the value function in (48) l times for each state $X_i \in \mathscr{X}_i$ until the convergence happens. Thus, in order to get the value functions, the value functions need to be calculated in total of $l|\mathscr{X}_i|$ times for terminal i and $l|\hat{\mathscr{X}_s}|$ times for server s. Given the value functions, the control actions for each state $X_i \in \mathscr{X}_i$ ($\hat{X}_s \in \hat{\mathscr{X}_s}$) can now be calculated from (49).

3.7 Stabilizing control policies

In this section, we analyse the performance of our dynamic control policy. We show that the performance of the dynamic policy is better than the performance of the frame based and stationary policy presented in [10], [12]. In addition, we prove that the frame based policy, that was proposed to provide performance better than the stationary policy in [10], [12], does not provide bound for the average queue length.

3.7.1 *K*-step Lyapunov drift

Our stability analysis relies on Lyapunov drift that specifies a sufficient condition for the stability of a system with queues. This method is used to prove the stability of different policies in several publications, such as [10], [12], [14], [38], [39] and [40].

Lyapunov drift at terminal i

The maximum service rate at terminal i is given as

$$\mu_{\max_i}^* = \max\{\mu_i^{\max}, \mu_{i0}^{\max}\}, \tag{63}$$

where

$$\mu^*_{\max_i} = \max_{\{s\in\mathscr{S},\vec{H}_i\in\mathscr{H}_i\}} \mu_i(P_i) + \mu_{is}(P_{is}, h_{is}) \tag{64}$$

and $\mu_{i0}^{\max}$ is given in (54). Such a value exists because the arrival rates are bounded [10], [12], [14].

Consider the K-step dynamics of unfinished work at terminal i:

$$q_i(t_0+K) = q_i(t_0) + \sum_{n=t_0}^{t_0+K-1} a_i(n) - \sum_{n=t_0}^{t_0+K-1} [\acute{\mu}_i(n)^* + \sum_{s\in\mathscr{S}} b_{is}(n)\mu_{is}(n)^*]. \tag{65}$$

We can write (65) as

$$q_i(t_0+K) = y_i(t_0) + \sum_{n=t_0+1}^{t_0+K-1} a_i(n) - \sum_{n=t_0}^{t_0+K-1} [\acute{\mu}_i(n)^* + \sum_{s\in\mathscr{S}} b_{is}(n)\mu_{is}(n)^*], \tag{66}$$

where $y_i(t_0) = q_i(t_0) + a_i(t_0)$. By adding $a_i(t_0+K)$ on both sides of (195), we get

$$y_i(t_0+K) = y_i(t_0) + \sum_{n=t_0+1}^{t_0+K} a_i(n) - \sum_{n=t_0}^{t_0+K-1} [\acute{\mu}_i(n)^* + \sum_{s\in\mathscr{S}} b_{is}(n)\mu_{is}(n)^*], \tag{67}$$

where $y_i(t_0+K) = q_i(t_0+K) + a_i(t_0+K)$. Inserting $y_i = y_i(t_0)$, $\acute{\mu}_i^* + \mu_{is}^* = \frac{1}{K}\sum_{n=t_0}^{t_0+K-1} \acute{\mu}_i(n)^* + \sum_{s\in\mathscr{S}} b_{is}(n)\mu_{is}(n)^*$ and $a_i = \frac{1}{K}\sum_{n=t_0+1}^{t_0+K} a_i(n)$ into (67), we have

$$y_i(t_0+K) = y_i + Ka_i - K(\acute{\mu}_i^* + \mu_{is}^*). \tag{68}$$

Squaring both sides of (68), defining the Lyapunov function as $L(y_{\mathrm{T}}) = y_i^2$ and taking conditional expectations of the inequality given $y_i(t_0)$, the K-step Lyapunov drift is given as:

$$\mathbb{E}\{L[y_{\mathrm{T}}(t_0+K)] - L[y_{\mathrm{T}}(t_0)]\big|y_i(t_0)\} \leq K^2M - 2Ky_i(t_0)\times$$
$$\Big[\mathbb{E}\{\frac{1}{K}\sum_{n=t_0}^{t_0+K-1} \acute{\mu}_i(n)^* + \sum_{s\in\mathscr{S}} b_{is}(n)\mu_{is}(n)^*\big|y_i(t_0)\} - \mathbb{E}\{\frac{1}{K}\sum_{n=t_0+1}^{t_0+K} a_i(n)\big|y_i(t_0)\}\Big], \tag{69}$$

where

$$M = (\mu^*_{\max_i})^2 + (a_i^{\max})^2. \tag{70}$$

The above equation represents Lyapunov drift for any resource allocation policy that we can have for the system and it was first presented in [20].

One should also note that since $y_i(K) = q_i(K) + a_i(K)$, where $q_i(K)$ is given in (65), the policy that minimizes $\frac{1}{K+1}\sum_{n=0}^{K} y_i(n)$ also minimizes $\frac{1}{K+1}\sum_{n=0}^{K} q_i(n)$.

Lyapunov drift at server s

The maximum service rate of terminal i at server s is

$$\hat{\mu}_s^{\max_i} \triangleq \max_{\{i \in \mathscr{I}\}} \hat{\mu}_s^i(\hat{P}^{\max}). \tag{71}$$

The K-step dynamics of unfinished work at server s are given by

$$\hat{q}_s^i(t_0+K) = \hat{q}_s(t_0) + \sum_{n=t_0}^{t_0+K-1} \mu_{is}(n)^* - \sum_{n=t_0}^{t_0+K-1} \hat{\mu}_s^i(n), \tag{72}$$

that can be written as

$$\hat{y}_s^i(t_0+K) = \hat{y}_s^i(t_0) + \sum_{n=t_0+1}^{t_0+K} \mu_{is}(n)^* - \sum_{n=t_0}^{t_0+K-1} \hat{\mu}_s^i(n). \tag{73}$$

By defining Lyapunov function as $L(\hat{y}_{\mathrm{ST}}) = (\hat{y}_s^i)^2$, the K-step Lyapunov drift is then given as

$$\begin{aligned}\mathbb{E}\{L[\hat{y}_{\mathrm{ST}}(t_0+K)] - L[\hat{y}_{\mathrm{ST}}(t_0)] \big| \hat{y}_s^i(t_0)\} \leq K^2\hat{M} - 2K\hat{y}_s^i(t_0) \times \\ \Big[\mathbb{E}\{\frac{1}{K}\sum_{n=t_0}^{t_0+K-1} \hat{\mu}_s^i(n) \big| \hat{y}_s^i(t_0)\} - \mathbb{E}\{\frac{1}{K}\sum_{n=t_0+1}^{t_0+K} \mu_{is}(n)^* \big| \hat{y}_s^i(t_0)\}\Big],\end{aligned} \tag{74}$$

where $\hat{M}$ is given as

$$\hat{M} = (\hat{\mu}_s^{\max_i})^2 + (\mu_{is}^{\max})^2 \tag{75}$$

and $\mu_{is}^{\max}$ is defined in (40). Equation (74) represents the Lyapunov drift for any resource allocation policy yielding service rate $\hat{\mu}_s^i$ at server s.

Since $\hat{y}_s^i(K) = \hat{q}_s^i(K) + \mu_{is}(K)^*$, the policy that minimizes $\max_{i \in \mathscr{I}} \left\{ \frac{1}{K+1} \sum_{n=0}^{K} \hat{y}_s^i(n) \right\}$, also minimizes $\max_{i \in \mathscr{I}} \left\{ \frac{1}{K+1} \sum_{n=1}^{K} \hat{q}_s^i(n) \right\}$.

3.7.2 Randomized stationary policy

In order to support every point in the network stability region described in Section 3.5, it is sufficient to consider only the class of stationary, randomized policies that take control decisions based on the current channel states only and do not consider the current workloads. The randomized stationary policy was presented in [12] and it can be implemented only if the channel steady state probabilities and both the average long-term capacity and the internal arrival rates c_{is}^* are known in advance. The details on

the stability analysis and the implementation of a stationary policy can be found in [10], [12]. Here, we compare the stationary policy and our dynamic control policy to each other.

The average arrival rates of each terminal i and the average arrival rates of each terminal i at servers s are assumed to be strictly inside the network stability region Λ, so that $\lambda_i + \theta \in \Lambda$ and $\lambda_{is}^{\text{ts}} + \theta \in \Lambda$. As the the stationary policy does not base its decisions on the current queue workloads, for every $t_0 \in \{0, ..., K-1\}$ it can be designed to provide [10], [12]

$$\mathbb{E}\{\frac{1}{K}\sum_{n=t_0}^{t_0+K-1} \acute{\mu}_i(n)^* + \sum_{s\in\mathscr{S}} b_{is}(n)\mu_{is}(n)^* \big| y_i(t_0)\} - \mathbb{E}\{\frac{1}{K}\sum_{n=t_0+1}^{t_0+K} a_i(n)\big|y_i(t_0)\} \geq \frac{2\theta}{3} \quad (76)$$

for each terminal i and

$$\mathbb{E}\{\frac{1}{K}\sum_{n=t_0}^{t_0+K-1} \hat{\mu}_s^i(n)\big|\hat{y}_s^i(t_0)\} - \mathbb{E}\{\frac{1}{K}\sum_{n=t_0+1}^{t_0+K} \mu_{is}(n)^*\big|\hat{y}_s^i(t_0)\} \geq \frac{2\theta}{3} \quad (77)$$

for each terminal i at server s. If a terminal or a server does not have enough (or any) data to process or send over, null bits are delivered.

Inserting (76) and (77) into right hand side of (69) and (74), respectively, the queuing bounds for the stationary policy can be given as [10], [12]

$$\limsup_{n\to\infty} \frac{1}{n+1}\sum_{\eta=0}^{n} \mathbb{E}\{q_i(\eta)\} \leq \limsup_{n\to\infty} \frac{1}{n+1}\sum_{\eta=0}^{n} \mathbb{E}\{y_i(\eta)\} \leq \frac{3KM}{4\theta} \quad (78)$$

for all $i \in \mathscr{I}$ and

$$\limsup_{n\to\infty} \frac{1}{n+1}\sum_{\eta=0}^{n} \mathbb{E}\{\hat{q}_s^i(\eta)\} \leq \limsup_{n\to\infty} \frac{1}{n+1}\sum_{\eta=0}^{n} \mathbb{E}\{\hat{y}_s^i(\eta)\} \leq \frac{3K\hat{M}}{4\theta} \quad (79)$$

for all $i \in \mathscr{I}$ and $s \in \mathscr{S}$.

3.7.3 Frame based policy

Frame based policy presented in [10], [12] works like the dynamic policy, but updates the backlog information every K frames. Given (37) and (50), the frame based policy is designed to maximize

$$\frac{1}{\hat{J}K}\sum_{j=0}^{\hat{J}-1}\sum_{n=jK}^{(j+1)K-1} \mathbb{E}\Big\{[y_i(jK) - \hat{q}_s^i(jK)]\frac{\sum_{s\in\mathscr{S}} b_{is}(n)\mu_{is}(n)^*}{\mu_{is}^{\max}} + y_i(jK)[\acute{\mu}_i(n)^* + \sum_{s\in\mathscr{S}} b_{is}(n)\mu_{is}(n)^*]\Big\} \quad (80)$$

at each terminal i and

$$\frac{1}{\hat{f}K}\sum_{j=0}^{\hat{f}-1}\sum_{n=jK}^{(j+1)K-1}\mathbb{E}\{\sum_{i\in\mathscr{I}}\hat{y}_s^i(jK)\hat{\mu}_s^i(n)\} \tag{81}$$

at each terminal i at server s. The frame based policy was proposed to stabilize the network and to provide better performance than the stationary policy in [10], [12].

Theorem 1. *In order for a policy to provide better performance than the stationary policy and to satisfy the bounds for average queue lengths in (78) and (79), it should be designed to maximize*

$$y_i(t_0)\Big[\mathbb{E}\{\frac{1}{K}\sum_{n=t_0}^{t_0+K-1}\acute{\mu}_i(n)^*+\sum_{s\in\mathscr{S}}b_{is}(n)\mu_{is}(n)^*\big|y_i(t_0)\}-\mathbb{E}\{\frac{1}{K}\sum_{n=t_0+1}^{t_0+K}a_i(n)\big|y_i(t_0)\}\Big]>0 \tag{82}$$

and

$$\hat{y}_s^i(t_0)\Big[\mathbb{E}\{\frac{1}{K}\sum_{n=t_0}^{t_0+K-1}\hat{\mu}_s^i(n)\big|\hat{y}_s^i(t_0)\}-\mathbb{E}\{\frac{1}{K}\sum_{n=t_0}^{t_0+K-1}\mu_{is}(n)^*\big|\hat{y}_s^i(t_0)\}\Big]>0 \tag{83}$$

on the right hand sides of (69) and (74) for each terminal i and server s and for every $t_0\in\{0,....,K-1\}$.

Proof. Rewriting (74) as

$$\begin{aligned}\mathbb{E}\{L[\hat{y}_{\mathrm{ST}}(t_0+(j+1)K)]-L[\hat{y}_{\mathrm{ST}}(t_0+jK)]\big|\hat{y}_s^i(t_0+jK)\}\le K^2\hat{M}-2K\hat{y}_s^i(t_0+jK)\times\\ \Big[\mathbb{E}\{\frac{1}{K}\sum_{n=t_0+jK}^{t_0+(j+1)K-1}\hat{\mu}_s^i(n)\big|\hat{y}_s^i(t_0+jK)\}-\mathbb{E}\{\frac{1}{K}\sum_{n=t_0+jK+1}^{t_0+(j+1)K}\mu_{is}(n)^*\big|\hat{y}_s^i(t_0+jK)\}\Big]\end{aligned} \tag{84}$$

and summing (84) over j and t_0 from $j=0$ to $j=\hat{f}-1$ and from $t_0=0$ to $t_0=K-1$, we get

$$\begin{aligned}&2K\sum_{t_0=0}^{K-1}\sum_{j=0}^{\hat{f}-1}\hat{y}_s^i(t_0+jK)\Big[\mathbb{E}\{\frac{1}{K}\sum_{n=t_0+jK}^{t_0+(j+1)K-1}\hat{\mu}_s^i(n)\big|\hat{y}_s^i(t_0+jK)\}-\\ &\mathbb{E}\{\frac{1}{K}\sum_{n=t_0+jK+1}^{t_0+(j+1)K}\mu_{is}(n)^*\big|\hat{y}_s^i(t_0+jK)\}\Big]\le K^3\hat{M}\hat{f}-\\ &\qquad\sum_{t_0=0}^{K-1}\sum_{j=0}^{\hat{f}-1}\mathbb{E}\{L[\hat{y}_{\mathrm{ST}}(t_0+(j+1)K)]-L[\hat{y}_{\mathrm{ST}}(t_0+jK)]\big|\hat{y}_s^i(t_0+jK)\}.\end{aligned} \tag{85}$$

If there exists a positive number ρ_i, such that

$$\Big[\mathbb{E}\{\frac{1}{K}\sum_{n=t_0+jK}^{t_0+(j+1)K-1}\hat{\mu}_s^i(n)\big|\hat{y}_s^i(t_0+jK)\}-\mathbb{E}\{\frac{1}{K}\sum_{n=t_0+jK+1}^{t_0+(j+1)K}\mu_{is}(n)^*\big|\hat{y}_s^i(t_0+jK)\}\Big]\ge\rho_i \tag{86}$$

for every $t_0 \in \{0,....,K-1\}$ and $j \in \{0,....,\hat{J}\}$, we can insert ρ_i on the left hand side of (85). Now, (85) can be rewritten as

$$2K\rho_i \sum_{t_0=0}^{K-1}\sum_{j=0}^{\hat{J}-1} \hat{y}_s^i(t_0+jK) \leq K^3\hat{M}\hat{J} - \sum_{t_0=0}^{K-1}\sum_{j=0}^{\hat{J}-1} \mathbb{E}\{L[\hat{y}_{\mathrm{ST}}(t_0+(j+1)K)] - L[\hat{y}_{\mathrm{ST}}(t_0+jK)]\big|\hat{y}_s^i(t_0+jK)\}. \quad (87)$$

Taking expectations over the distribution of $\hat{y}_s^i(t_0+jK)$, dividing by $\hat{J}$ and K^2 and using the non-negativity of the Lyapunov function, we get

$$2\rho_i \frac{1}{\hat{J}K}\sum_{t_0=0}^{K-1}\sum_{j=0}^{\hat{J}-1} \mathbb{E}\{\hat{y}_s^i(t_0+jK)\} \leq K\hat{M} + \frac{1}{\hat{J}K^2}\sum_{t_0=0}^{K-1} \mathbb{E}\{L[\hat{y}_{\mathrm{ST}}(t_0)]\}. \quad (88)$$

Using lim sup on the above inequality, when $\hat{J} \to \infty$ yields the performance bound:

$$\limsup_{n\to\infty} \frac{1}{n+1}\sum_{\eta=0}^{n} \mathbb{E}\{\hat{y}_s^i(\eta)\} \leq \frac{K\hat{M}}{2\rho_i}. \quad (89)$$

As the derivation of the above bound is similar for (69), it is omitted for brevity.

It is now easy to see that in order for a policy to outperform the stationary policy and to minimize the long-term average queue length, the policy must be designed to maximize (82) and (83) at each terminal i and server s and for every $t_0 \in \{0,....,K-1\}$. □

Given (80) and (81), it is easy to see that the frame based policy does not satisfy Theorem 1. In addition, as the frame based policy is designed to maximize the sum over all the terminals in (81) and updates queue lengths only every K frame, it is possible that all/most of the resources are allocated to a single terminal. In such a case,

$$\sum_{i\in\mathscr{I}}\Big[\mathbb{E}\{\frac{1}{K}\sum_{n=jK}^{(j+1)K-1}\hat{\mu}_s^i(n)\big|\hat{y}_s^i(t_0)\} - \mathbb{E}\{\frac{1}{K}\sum_{n=jK}^{(j+1)K-1}\mu_{is}(n)^*\big|\hat{y}_s^i(t_0)\}\Big] > 0, \quad (90)$$

but for some individual terminals

$$\Big[\mathbb{E}\{\frac{1}{K}\sum_{n=jK}^{(j+1)K-1}\hat{\mu}_s^i(n)\big|\hat{y}_s^i(t_0)\} - \mathbb{E}\{\frac{1}{K}\sum_{n=jK}^{(j+1)K-1}\mu_{is}(n)^*\big|\hat{y}_s^i(t_0)\}\Big] < 0. \quad (91)$$

Thus, the frame based policy does not provide better performance than the stationary policy and it cannot provide bound for the average queue length as given in (89).

3.7.4 Dynamic control policy

In this section, we show that our dynamic control policy offers better performance than the stationary and frame based policy and provides bounds on average delays at each terminal i and server s.

Theorem 2. *Dynamic policy stabilizes the network and the performance of the dynamic policy is better than the performance of the frame based and the randomized stationary algorithms.*

Proof. Dynamic control policy is designed to maximize (37) at each terminal i and (50) at each servers s. We rewrite (37) and (50), as

$$\frac{1}{\hat{J}K}\sum_{j=0}^{\hat{J}-1}\sum_{n=jK}^{(j+1)K-1}\mathbb{E}\Big\{[y_i(n)-\hat{q}_s^i(n)]\frac{\sum_{s\in\mathscr{S}}b_{is}(n)\mu_{is}(n)^*}{\mu_{is}^{\max}}+ y_i(n)[\acute{\mu}_i(n)^*+\sum_{s\in\mathscr{S}}b_{is}(n)\mu_{is}(n)^*]\Big\} \tag{92}$$

and

$$\frac{1}{\hat{J}K}\sum_{j=0}^{\hat{J}-1}\sum_{n=jK}^{(j+1)K-1}\mathbb{E}\Big\{\sum_{i\in\mathscr{I}}\hat{y}_s^i(n)\hat{\mu}_s^i(n)\Big\}, \tag{93}$$

respectively. Noting that $y_i(n)=y_i(jK)+\sum_{\eta=jK+1}^{n}a_i(\eta)-\sum_{\eta=jK}^{n-1}[\acute{\mu}_i(\eta)^*+\sum_{s\in\mathscr{S}}b_{is}(\eta)\times\mu_{is}(\eta)^*]$ and $\hat{y}_s^i(n)=\hat{y}_s^i(jK)+\sum_{\eta=jK+1}^{n}\mu_{is}(\eta)^*-\sum_{\eta=jK}^{n-1}\hat{\mu}_s^i(\eta)$, we see that the dynamic policy maximizes

$$\begin{aligned}
&\frac{1}{\hat{J}K}\Bigg[\mathbb{E}\Big\{\sum_{n=0}^{K-1}[y_i(n)-\hat{q}_s^i(n)]\frac{\sum_{s\in\mathscr{S}}b_{is}(n)\mu_{is}(n)^*}{\mu_{is}^{\max}}+y_i(0)[\acute{\mu}_i(n)^*+\sum_{s\in\mathscr{S}}b_{is}(n)\mu_{is}(n)^*]+\\
&\quad[\acute{\mu}_i(n)^*+\sum_{s\in\mathscr{S}}b_{is}(n)\mu_{is}(n)^*]\Big[\sum_{\eta=1}^{n}a_i(\eta)-\sum_{\eta=0}^{n-1}[\acute{\mu}_i(\eta)^*+\sum_{s\in\mathscr{S}}b_{is}(\eta)\mu_{is}(\eta)^*]\Big]\Big\}+\ldots+\\
&\mathbb{E}\Big\{\sum_{n=(\hat{J}-1)K}^{\hat{J}K-1}[y_i(n)-\hat{q}_s^i(n)]\frac{\sum_{s\in\mathscr{S}}b_{is}(n)\mu_{is}(n)^*}{\mu_{is}^{\max}}+y_i((\hat{J}-1)K)[\acute{\mu}_i(n)^*+\sum_{s\in\mathscr{S}}b_{is}(n)\mu_{is}(n)^*]+\\
&[\acute{\mu}_i(n)^*+\sum_{s\in\mathscr{S}}b_{is}(n)\mu_{is}(n)^*]\Big[\sum_{\eta=(\hat{J}-1)K+1}^{n}a_i(\eta)-\sum_{\eta=(\hat{J}-1)K}^{n-1}[\acute{\mu}_i(\eta)^*+\sum_{s\in\mathscr{S}}b_{is}(\eta)\mu_{is}(\eta)^*]\Big]\Big\}\Bigg]
\end{aligned} \tag{94}$$

at each terminal i and

$$\frac{1}{\hat{J}K}\Bigg[\mathbb{E}\Big\{\sum_{n=0}^{K-1}\sum_{i\in\mathscr{I}}\hat{y}_s^i(0)\hat{\mu}_s^i(n)+\hat{\mu}_s^i(n)\Big[\sum_{\eta=1}^{n}\mu_{is}(\eta)^*-\sum_{\eta=0}^{n-1}\hat{\mu}_s^i(\eta)\Big]\Big\}+ \tag{95}$$

$$\mathbb{E}\Big\{\sum_{n=K}^{2K-1}\sum_{i\in\mathscr{I}}\hat{y}_s^i(K)\hat{\mu}_s^i(n)+\hat{\mu}_s^i(n)\Big[\sum_{\eta=K+1}^{n}\mu_{is}(\eta)^*-\sum_{\eta=K}^{n-1}\hat{\mu}_s^i(\eta)\Big]\Big\}+\ldots+$$

$$\mathbb{E}\Big\{\sum_{n=(\hat{J}-1)K}^{\hat{J}K-1}\sum_{i\in\mathscr{I}}\hat{y}_s^i((\hat{J}-1)K)\hat{\mu}_s^i(n)+\hat{\mu}_s^i(n)\Big[\sum_{\eta=(\hat{J}-1)K+1}^{n}\mu_{is}(\eta)^*-\sum_{\eta=(\hat{J}-1)K}^{n-1}\hat{\mu}_s^i(\eta)\Big]\Big\}\Bigg]$$

at each servers s.

By comparing (80) to (94) and (81) to (95) it can also be seen that unlike the frame based policy, our dynamic policy allocates more resources to a terminal with the longest queue so that in every time frame $\acute{\mu}_i(n)^*+\sum_{s\in\mathscr{S}}b_{is}(n)\mu_{is}(n)^*-a_i(n)$ and $\hat{\mu}_s^i(n)-\mu_{is}(n)^*$ are maximized for each queue at terminal i and server s. Thus, our dynamic policy maximizes the right hand sides of (69) and (74) for all t_0 and minimizes the long-term average queues both at the terminals and at the servers. The dynamic policy stabilizes the network and its performance is better that the performance of the randomized stationary or frame based policy.

□

3.8 Performance evaluation

For illustration purposes, we have evaluated the performance of the dynamic control policy with simulations. The performance of the optimal dynamic transmission policy is illustrated in the presence of time varying workloads and uncertain channels both for CN and PC network as well as for CWN. It is shown that by adapting to the changes in network conditions, our control policy mitigates the effect of PSP and SSP cognitive networks on each other. The simulations support our stability analysis presented in Sections 3.5 and 3.7, and are implemented using Matlab.

3.8.1 Experiment setup

Due to the complexity of the problem, we set $|\mathscr{I}|=|\mathscr{S}|=2$. Although the simulations are run only for a small system, we would like to point out that the stability analysis is valid for any size of the system in Section 3.7. The channel process is generated according to a Markov chain and state transition matrix for the channel between terminal

i and the hosting servers in the CWN is given as

$$\mathbf{T} = \begin{bmatrix} T_{11} & T_{12} & T_{13} & T_{14} \\ T_{21} & T_{22} & T_{23} & T_{24} \\ T_{31} & T_{32} & T_{33} & T_{34} \\ T_{41} & T_{42} & T_{43} & T_{44} \end{bmatrix} = \begin{bmatrix} 0.3 & 0.5 & 0.2 & 0 \\ 0.1 & 0.6 & 0.2 & 0.1 \\ 0.1 & 0.3 & 0.5 & 0.1 \\ 0 & 0.1 & 0.25 & 0.65 \end{bmatrix}, \tag{96}$$

where T_{kl} is the probability of transitioning from channel state k to l, and the corresponding stationary probabilities $p\{\vec{H}_i = [|h_{11}|^2, |h_{12}|^2]\}$ are given as $p\{\vec{H}_i = [10, 10]\} = 0.1$, $p\{\vec{H}_i = [10, 20]\} = 0.4$, $p\{\vec{H}_i = [20, 10]\} = 0.3$, $p\{\vec{H}_i = [20, 20]\} = 0.2$.

For the SSP cognitive network, the probability that the channels between terminal i and the servers are available for communication is $p_H^{\mathrm{S}} = 0.9$ or $p_H^{\mathrm{S}} = 0.7$, that is given in (17). The stationary probabilities are then given as $p\{\vec{H}_i = [10, 10]\} = 0.09$, $p\{\vec{H}_i = [10, 20]\} = 0.36$, $p\{\vec{H}_i = [20, 10]\} = 0.27$, $p\{\vec{H}_i = [20, 20]\} = 0.18$, $p\{\vec{H}_i = [0, 0]\} = 0.1$ or $p\{\vec{H}_i = [10, 10]\} = 0.07$, $p\{\vec{H}_i = [10, 20]\} = 0.28$, $p\{\vec{H}_i = [20, 10]\} = 0.21$, $p\{\vec{H}_i = [20, 20]\} = 0.14$, $p\{\vec{H}_i = [0, 0]\} = 0.3$. The probability that PU returns to the given channel in is $p_{\mathrm{return}}^{\mathrm{P}} = 0.05$.

In the PC network, where the overall resources include both cognitive and conventional links, we assume that the channel between terminal i and server 1 is cognitive and the channel between terminal i and server 2 is a non-cognitive channel. The probability that the channel between terminal i and server 1 is available for communication is $p_H^{\mathrm{S}} = 0.9$ or $p_H^{\mathrm{S}} = 0.7$. The stationary probabilities are given as $p\{\vec{H}_i = [10, 10]\} = 0.09$, $p\{\vec{H}_i = [10, 20]\} = 0.36$, $p\{\vec{H}_i = [20, 10]\} = 0.27$, $p\{\vec{H}_i = [20, 20]\} = 0.18$, $p\{\vec{H}_i = [0, 10]\} = 0.05$, $p\{\vec{H}_i = [0, 20]\} = 0.05$ or $p\{\vec{H}_i = [10, 10]\} = 0.07$, $p\{\vec{H}_i = [10, 20]\} = 0.28$, $p\{\vec{H}_i = [20, 10]\} = 0.21$, $p\{\vec{H}_i = [20, 20]\} = 0.14$, $p\{\vec{H}_i = [0, 10]\} = 0.15$, $p\{\vec{H}_i = [0, 20]\} = 0.15$. The probability that PU returns to the given channel between terminal i and server 2 is $p_{\mathrm{return}}^{\mathrm{P}} = 0.05$.

For a Poisson process, the second moment of arrivals in each frame is finite [12]. Thus, each terminal is assumed to receive requests from applications according to a Poisson process at an average rate of λ_i. In the simulations, λ_i takes values between 1 to 8 requests/frame, and $\lambda_1 = \lambda_2$. The maximum available power at each terminal is $P_{\max} = 4W$. Let $\alpha_i = 0.6$ in (21), the discount factor $\alpha = 0.7$ in (46) and $\alpha_{is} = 100$ in (22). The Lagrangian multiplier in (44) is fixed to $\beta_i = 1$. The long-term average sum power, sum delay, and sum throughput are calculated over $N = 20000$ frames.

Each CPU is assumed to follow a quadratic power-frequency relationship. Specifically, CPU is assumed to have a discrete set of frequency options in the interval [1.6GHz,

..., 2.6GHz] at increments of 0.2 GHz and the corresponding power consumption (in watts) at frequency f is given by $\hat{P}_{\min} + \theta(f - 1.6GHz)^2$ where $\hat{P}_{\min} = 10W$ and $\theta = 10W/(GHz)^2$. Thus, the CPU power consumption at the highest frequency is $20W$. At each utilization level f, the maximum supportable service rate $\mu_{is}^{\max}(f)$ is given in (25), where $\hat{\alpha}_s = 0.4$. Thus, on average, a server running at the minimum (maximum) speed can process 25 (50) requests per frame.

3.8.2 Numerical Results and Discussions

In the figures we have used the following notations: 'CWN' - conventional wireless network, 'CN' - cognitive network, 'PC' - partial cognitive network, 'T'- terminals, 'S'- servers, 'TS'- transmission from terminals to servers and 'NW' - entire network. In addition, '10%' and '30%' represent the probabilities that the channel between terminal i and server 1 is not available for communication.

The average sum service rates at the terminals (T) and the average sum rates from terminals to servers (TS) are plotted as a function of $\lambda_1 + \lambda_2$ for both the CWN and the PC network in Fig. 4. It can be seen in the figure, that the average sum service rates at the terminals both in the CWN and the PC network equal $\lambda_1 + \lambda_2$. In the CWN, almost all application requests are forwarded to be processed at the servers. In the PC network, the effect of PSP and SSP cognitive networks on each other is mitigated by processing considerably more requests at the terminals. If the channel between the terminal and server 1 is not available for communication, and if the channel between the terminal and server 2 is bad, the more requests are processed at the terminal, especially when the arrival rates are low. However, it also can be seen in Fig. 4, that the number of requests forwarded to the servers gets higher with the increase of $\lambda_1 + \lambda_2$. This is due to the smaller processing capabilities at the terminals than at the servers.

The average sum delays at the terminals (T) and the average sum delays over the entire network (NW) are plotted as a function of $\lambda_1 + \lambda_2$ for both CWN and PC network in Fig. 5. It can be seen, that for the given system parameters the processing delay at the servers decreases as $\lambda_1 + \lambda_2$ increases, when $\lambda_1 + \lambda_2 < 9$. This is because, at low arrival rates, the queues at the servers are short. Thus, in order to maximize (50), it is more advantageous to delay some of the requests in order to achieve energy savings at the server. When $\lambda_1 + \lambda_2$ is large, there is no much processing delay at the servers, because high arrival rates from the terminals encourage servers to empty their queues by increasing their processing capabilities. Due to the uncertain availability and reliability

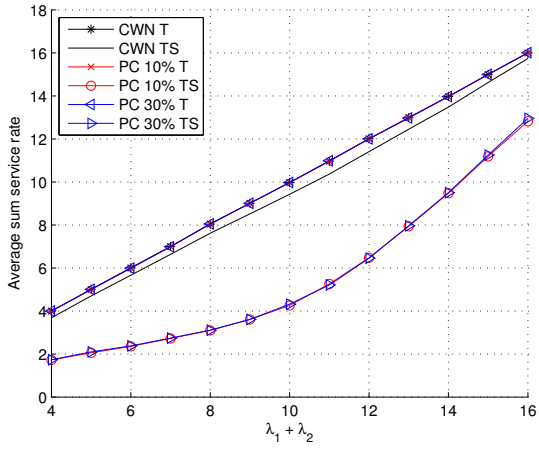

Fig. 4. Average sum rates vs. $\lambda_1 + \lambda_2$ for both CWN and PC network.

of the channel between the terminals and server 1 in the PC network, the delay at server 1 is longer in the PC network than in the CWN. Thus, also the overall network delay in the PC network is longer than that of the CWN. It can also be seen, that the overall network delay in the PC 30% network is a bit shorter than in the PC 10% network. This is due to the fact that, even if the channel between the terminals and server 1 is not available for communication, the channel between the terminals and server 2 is. In addition, the probability that the transmission over the given channel between terminal i and server 1 fails is smaller in the PC 30% network than in the PC 10% network, since $p(I = 1)p(I_r = 0) = 0.7 \times 0.5 = 0.35$ and $p(I = 1)p(I_r = 0) = 0.9 \times 0.5 = 0.45$.

The average sum power consumptions both at terminals (T) and servers (S) are plotted as a function of $\lambda_1 + \lambda_2$ for CWN and PC network in Fig. 6. As most of the requests are processed at the servers in the CWN, the power consumption at the servers is significantly higher than the power consumption at the terminals. Due to the uncertain availability and reliability of the channel between the terminals and server 1 in the PC network, terminals consume more power in the PC network than in the CWN. If the channel between terminal i and server 1 is not available for communication, or if the channel between terminal i and server 2 is bad, it is more advantageous in terms of saving the transmission power to process more requests at the terminal. For the given range of $\lambda_1 + \lambda_2$, the power consumption at the servers in the CWN is smaller when $\lambda_1 + \lambda_2 \geq 13$ than when $7 < \lambda_1 + \lambda_2 < 13$. As mentioned earlier in this chapter, the

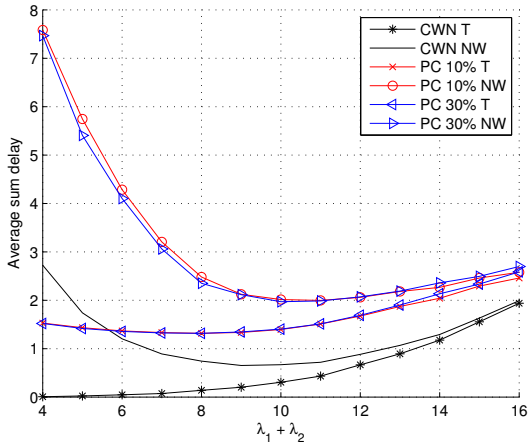

Fig. 5. Average sum delays vs. $\lambda_1 + \lambda_2$ for both CWN and PC network.

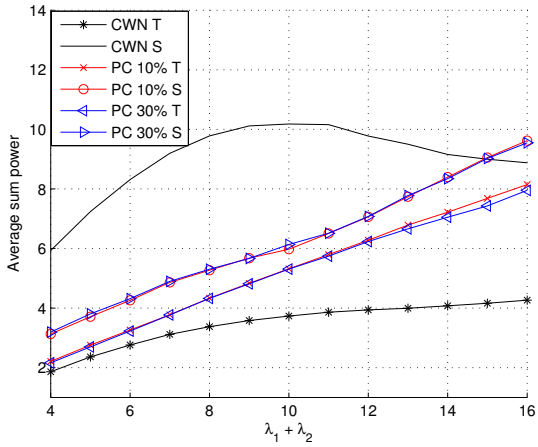

Fig. 6. Average sum powers vs. $\lambda_1 + \lambda_2$ for both CWN and PC network.

server consumes at least $\hat{P}_{\min}$ even to process only a small amount of data. Thus, the active servers do not necessarily always process the maximum number of requests that could be processed with the used power, when $7 < \lambda_1 + \lambda_2 < 13$. If $\lambda_1 + \lambda_2$ is large, the used power can be better utilized in every frame, and more data can be processed with the lower power consumption. It can also be seen, that the average sum power in the PC 30% network is very close to the average sum power in the PC 10%

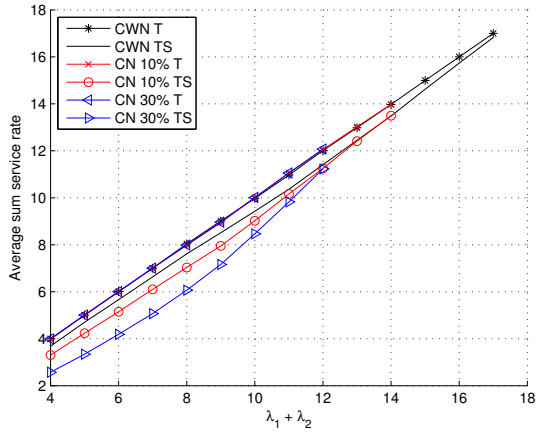

Fig. 7. Average sum rates vs. $\lambda_1 + \lambda_2$ for both CWN and SSP cognitive network.

network. This is because the channel between the terminal and server 2 is non-cognitive and the probability that the transmission over the given channel between terminal i and server 1 fails is smaller in the PC 30% network than in the PC 10% network, i.e., $p(I = 1)p(I_r = 0) = 0.7 \times 0.5 = 0.35$ and $p(I = 1)p(I_r = 0) = 0.9 \times 0.5 = 0.45$. In addition, due to the uncertain link availability and reliability between server 1 and the terminals, server 1 does not receive as many requests as server 2. However, as severs consumes at least $\hat{P}_{\min}$ to process any amount of data, server 1 consumes almost equal amount of power as server 2. For the given arrival rates there is not enough requests to fully exploit the available power at server 1 and that is why the sum power consumption at the servers increases for all $\lambda_1 + \lambda_2$.

The average sum service rates at the terminals (T) and the average sum rates from terminals to servers (TS) are plotted as a function of $\lambda_1 + \lambda_2$ for both CWN and SSP cognitive network (CN) in Fig. 7. It can be seen that the average sum service rates at the terminals equal $\lambda_1 + \lambda_2$ for both networks supporting our stability analysis in Sections 3.5 and 3.7. However, due to the different network stability regions, the maximum supportable arrival rates in cognitive wireless networks is smaller than in the CWN. It can be seen, that the probability to process requests at the terminals is slightly higher in the CN than in the CWN, when arrival rates are low. This is due to the uncertain channel availability and reliability between the terminals and the servers. However, for high arrival rates, most of the requests are processed at the server only also in cognitive

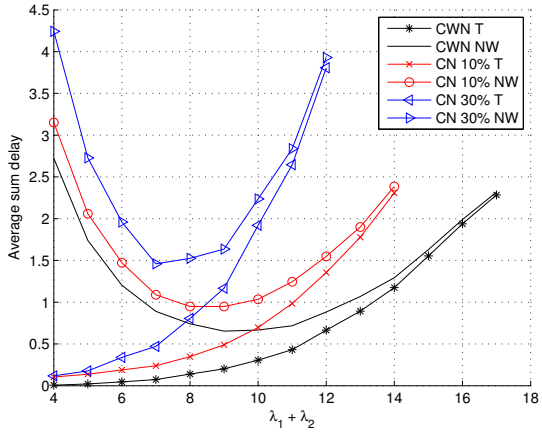

Fig. 8. Average sum delays vs. $\lambda_1 + \lambda_2$ for both CWN and SSP cognitive network.

wireless network. For high arrival rates, it is more beneficial in terms of decreasing the transmission power and the delay to forward the application requests to the servers.

The average sum delays at the terminals (T) and average sum delays over the entire network (NW) are plotted as a function of $\lambda_1 + \lambda_2$ for both CWN and CN in Fig. 8. Due to the uncertain channel availability and reliability between the terminals and the servers, the delay in the CN is significantly longer than in the CWN. It can also be seen, that the processing delay at the servers decreases as $\lambda_1 + \lambda_2$ increases, when $\lambda_1 + \lambda_2$ is small. This is because, at low arrival rates, the queues at the servers are short. Thus, it is more advantageous to delay some of the requests in order to achieve energy savings at the server. When $\lambda_1 + \lambda_2$ is large, there is not much processing delay at the servers, because high arrival rates from the terminals encourage servers to empty their queues by increasing the capability to process the requests.

The average sum power consumptions both at the terminals (T) and the servers (S) are plotted as a function of $\lambda_1 + \lambda_2$ for CWN and CN in Fig. 9. It can be seen in the figure, that in the cognitive network our policy consumes approximately 10% or 30% less power at the servers than the policy consumes in the CWN. That is due to the uncertain channel availability between the terminals and the servers in the cognitive wireless network. It can also be seen, that the power consumption at the terminals in the cognitive network is slightly smaller or equal to power consumption in the CWN. The delay in the CN is significantly longer than in the CWN, since the terminals delay

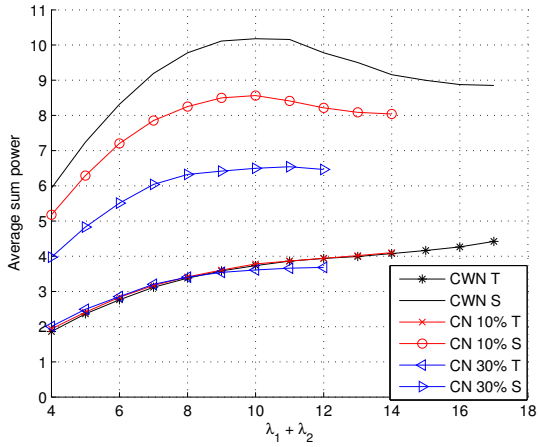

Fig. 9. Average sum powers vs. $\lambda_1 + \lambda_2$ for both CWN and SSP cognitive network.

they requests waiting for the available channels or better channel conditions. Thus, the average power consumption at the terminals in the cognitive network is slightly smaller than in the CWN.

3.9 Chapter summary

We have considered a virtualized data center (computing cloud) consisting of a set of servers hosting a number of mobile terminals (a mobile cloud), and have studied the problem of resource allocation in the presence of time varying workloads and uncertain channels. The channel uncertainty was either due to fading and/or uncertain link availability and reliability in PSP/SSP cognitive networks. We have presented a unified VDC model for both cognitive and conventional wireless networks. For this unified model, we have designed a new dynamic resource allocation policy that maximizes the jointly utility of the long-term average throughput and minimizes the energy consumption, both at terminals and servers, while maintaining network stability. We have characterized the unified network stability region for both SSP and PSP cognitive networks as well as for the CWN, and presented a new unified stability analysis for the three networks. The proposed dynamic policy was shown to support every point on the network stability region using the Lyapunov drift theory. Performance evaluation has been carried out in order to compare the performance of dynamic control

policy in the CWN with the performance of dynamic policy in the SSP/PSP cognitive wireless networks, and to validate the theoretical analysis of the chapter. The results have shown that by adapting to the changes in network conditions, our dynamic policy can mitigate the impact of PSP and SSP cognitive networks on each other. We believe that the presented approach can be used as a performance benchmark and lays the foundation for future solutions of different simplified resource allocation schemes in VDC computing clouds. The major contributions of this chapter can be summarized as follows:

- A comprehensive unified model of the virtualized data center (computing could) for both PSP and SSP cognitive networks as well as for CWNs is developed.
- The model decouples performance analysis of PSP and SSP cognitive networks although their operations are interdependent.
- The mutual impact of PSP and SSP cognitive networks is mitigated by appropriate adaptation of the access control parameters in the network.
- New unified dynamic control policy is introduced.
- Unified stability region for PSP and SSP cognitive networks and CWNs is characterized.
- Unified stability analysis for both PSP and SSP cognitive networks as well as for CWNs is presented.
- It is shown that the performance of the frame based policy is not better than the performance of the stationary randomized policy and that the frame based policy does not provide bound for the average queue length.
- Using the Lyapunov drift theory, it is shown that our dynamic policy supports every point in the network stability region and that its performance policy is better than the performance of the stationary randomized policy proposed in [10], [12].

4 The stability of cooperative cognitive wireless networks

In this chapter, we propose a number of cooperative strategies in cognitive wireless networks that generate additional capacity. We consider a mobile cloud with cooperative communication and queueing, and study the problem of optimal resource allocation in the presence of uncertain channels. As in Chapter 3, the channel uncertainty is due to the fading and/or uncertain link availability and reliability both in PSP and SSP CNs.

First, a detailed discussion on the unified system model for both PSP and SSP cognitive networks and for CWN is presented. For this model, we formulate a unified optimization problem, where the goal is to assign resources dynamically in reaction to changes in workloads and channel conditions in order to maximize the long-term average throughput of the system while providing bound on average delay. As the control problem is a constrained dynamic optimization problem, dynamic programming methods are used to provide an optimal dynamic control policy for both PSP and SSP cognitive networks as well as for CWN.

In order to compare the potential performances of the conventional non-cooperative (NC) and cooperative communication systems to each other, we establish the unified network stability regions of the cooperative and NC communication systems for both PSP and SSP cognitive networks and for CWN. In addition, the concept of Inter Technology Networking (InTeNet) is introduced to derive an upper bound for the stability region of the cooperative communication system.

We present a unified stability analysis for both PSP and SSP cognitive networks as well as for CWNs. We also show that our dynamic control policy minimizes the average maximum queue length over all terminals and stabilizes the network.

Finally, the simulation results are provided to compare the performance of the cooperative policy to the corresponding non-cooperative case and to support the theoretical analysis of this chapter.

This chapter is organized as follows. In Section 4.1, the motivation behind the research and the related work is presented. The formulation of the unified system model is described in Section 4.2. In Section 4.3, we model the unified optimization problem. The unified control problem is formulated as a MDP and solved using VIA in Section 4.4. In order to compare the potential performances of the NC and cooperative

communication systems to each other, the unified network stability regions are illustrated in Section 4.5. Section 4.6 contains the stability analysis and the simulation results are provided to validate our theoretical analysis in Section 4.7. Finally, some concluding remarks are offered in Section 4.8.

4.1 Motivation and related work

Here, the motivation behind the work presented in this chapter and the related work on this area are presented.Fading in wireless channels, the increasing demand for high data rate services, the limited battery energy available in wireless handsets and the changing user demands pose serious design challenges in wireless environments that need to be simultaneously taken into account when planning future wireless transmission policies in the CWNs. The problem becomes even more complex if the communication is organized within the SSP cognitive network, where additional disruptions to the channel are caused by unpredictable returns of users belonging to the PSP CN [85], [96]. It is critical to design efficient control algorithms to adapt to the changes in network conditions in order to achieve target delay and rate with minimum power consumption.

By using independently fading copies of a signal, diversity techniques can be used to mitigate the effects of fading and channel uncertainty. The advantages of Multiple-Input Multiple-Output (MIMO) systems have been widely acknowledged [97]. Although the use of multiple antennas is clearly advantageous at base stations in cellular system, it might be impractical in the uplink due to the size, cost and hardware limitations of a mobile unit.

The broadcast nature of wireless transmissions and work on MIMO systems motivate the use of cooperative communication to improve the performance of wireless networks with uncertain channels. User cooperation has been found to allow single-antenna mobiles in a multi-user environment to reap some of the benefits of MIMO systems by enabling them to share their antennas and form a virtual multiple-antenna transmitter [98], [99]. The proposed schemes in cooperative communication area have been shown to offer significant gains in several metrics such as diversity gains, capacity and power gains over direct communication and traditional relaying methods [99], [100], [101]. Interested readers are referred to an extensive survey in [102] on this topic.

A variety of techniques such as exploiting diversity, adaptive communication and power control, are used to combat fading and to meet even more stringent delay, power and throughput requirements. If the control decisions are based on both the channel

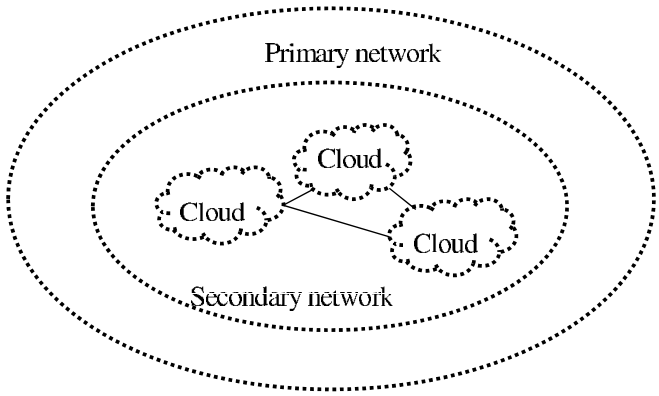

Fig. 10. A cognitive network.

state and queue length information, taking into account the changing user demands has been shown to be useful in providing higher throughput and shorter delay in the presence of time varying channels [11], [12]. Cooperative communication is especially attractive technique to provide additional reliability offering significant spatial diversity gains when used together with the above mentioned techniques.

Dynamic cooperation has been also considered in [38] and [103], where a dynamic control algorithm is a generalization of backpressure algorithms [18]. Our approach is different in the sense that it includes a number of different cooperative strategies, and it is one of the first attempts to provide a unified stability region for a cooperative network both in a CWN and a CN environment.

4.2 System model and assumptions

The cognitive network considered in this work is illustrated in Fig. 10. Each cloud is composed of an AP and a set of terminals with queues that are located within transmission range of each other. In a SSP CN, the AP is a cognitive router (CR) serving a cloud of mobile terminals [104]. In a cellular network, AP is a base station (BS) or a conventional AP in WLAN. The APs are then connected by a backhaul network to create the overall network.

In the literature, the term cloud has been used to refer to two different things. Mainly, the term cloud refers to a static hardware infrastructure, i.e., data center, that consists of a collection of virtualized servers and forms a large computing system that provides

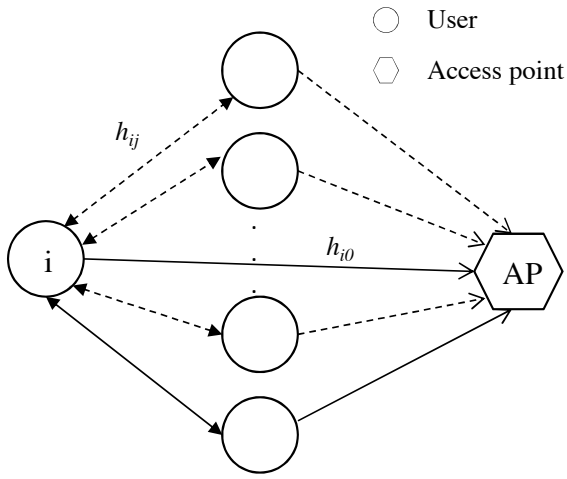

Fig. 11. Potential connections between user i and the AP inside the cloud.

users different kind of services like in Chapter 3. Alternatively, the term cloud can also refer to a set/cluster of terminals sharing a certain pool of resources. For example, an ad hoc mobile cloud consists of a group of mobile users that share the local resources in the ad hoc network with each other. In our case, users can borrow temporary channel from each other or collectively form a distributed MIMO system to repeat the transmission for a specific user.

Let $\mathscr{I}$ represent the set of terminals within a cloud and $|\mathscr{I}|$ denotes its cardinality. Due to the notational complexity to illustrate all possible connections between the terminals inside the cloud, only the connections between terminal i ($i = \{1,2,...,|\mathscr{I}|\}$) and the AP are illustrated in Fig. 11. From the set of potential connections between terminal i and the AP illustrated in Fig. 11, terminal i selects to connect with an AP either directly or via another terminal within the cloud. This is illustrated by the solid lines in Fig. 11.

Time is divided into time frames indexed by n. Application requests arrive to each terminal i according to a process $a_i(n)$ at the beginning of each frame n. The arrival processes $a_i(n)$ are stationary and ergodic with average rates λ_i requests/slot and the external arrivals $a_i(n)$ are bounded in their second moments every time slot and $\mathbb{E}\{[a_i(n)]^2\} \leq a_{max}^2$. Let $\vec{A}(n) = [a_1(n),...,a_{|\mathscr{I}|}(n)]$ denote the vector of these arrivals. For analysis purpose, we assume that the application requests are placed into infinite length transmission buffers.

4.2.1 Channel Model

We use $h_{i0}(n)$ to denote the channel state between terminal i and AP in frame n, and $h_{ij}(n)$ is the channel between terminal i and terminal j. Channel states are assumed to remain fixed during a frame and change from one frame to another according to a Markov chain [2]. Let vector $\vec{H}(n) = [|h_{10}(n)|^2, |h_{20}(n)|^2, ..., |h_{|\mathscr{I}|0}(n)|^2, |h_{12}(n)|^2, ... , |h_{1|\mathscr{I}|}(n)|^2, ..., |h_{|\mathscr{I}|1}(n)|^2, ... ,|h_{|\mathscr{I}|(|\mathscr{I}|-1)}(n)|^2]$ denote the channel gains in frame n that represents the cloud of resources shared among the users [3]. The vector $\vec{H}(n)$ is assumed to be stationary and ergodic and takes values on a finite state space $\mathscr{H}$.

If the channel is used within the CWN, the channel gain vector is given by $\vec{H}(n)$ in every frame n. Let π_H represent the steady state probability for the channel state $\vec{H}$ in the CWN.

When the channel is used within the CN, the equivalent channel gain vector $\vec{H}_e(n)$ has the following form:

$$\vec{H}_e(n) = \begin{cases} \vec{H}(n), & \text{with probability } p_H^{\mathrm{P}} \text{ for PSP CN or} \\ & \text{with probability } p_H^{\mathrm{S}} \text{ for SSP CN} \\ 0, & \text{with probability } p_0^{\mathrm{P}} \text{ for PSP CN or} \\ & \text{with probability } p_0^{\mathrm{S}} \text{ for SSP CN} \end{cases}$$

For the PSP cognitive network,

$$p_H^{\mathrm{P}} = (1 - p_1^{\mathrm{S}}) + p_1^{\mathrm{S}} p_{\mathrm{pd}} \tag{97}$$

and

$$p_0^{\mathrm{P}} = p_1^{\mathrm{S}}(1 - p_{\mathrm{pd}}). \tag{98}$$

Here, we assume that primary user (PU) transmits a preamble prior to message transmission to clear the channel in case that secondary user (SU) is using it (with probability p_1^{S}). Secondary user detects correctly that preamble and clears the channel with probability p_{pd}. Let p_1^{P} represent the probability that a PU is active and p_{id} is

[2]The finite state block fading Markov chain has been widely used to model the channel in the literature, e.g. [12], [15] and [20]. By using block fading model for the channel, we can dynamically generate artificial channel states that are analytically tractable and provide closed-form results. The assumption that the channels hold their states during a frame is an approximation, which is valid for systems, whose frames are short in comparison to the channel variation. In practice, channels may vary continuously.

[3]This definition differs slightly from the common understanding where the cloud represents a set of computing resources located in the Internet.

the probability that a SU detects the idling channel. In that case, in the SSP cognitive network,

$$p_H^{\mathrm{S}} = (1 - p_1^{\mathrm{P}}) p_{\mathrm{id}} \tag{99}$$

and the probability that the channel cannot be used is

$$p_0^{\mathrm{S}} = (1 - p_1^{\mathrm{P}})(1 - p_{\mathrm{id}}) + p_1^{\mathrm{P}}. \tag{100}$$

In other words, the channel gains are given by $\vec{H}(n)$, if the PU is not active and the SU detects the idling channel. The channel is not used, if the primary user is not active but the SU fails to detect the idling channel or the PU is active (probability p_1^{P}). The derivation of the probability 1 - p_1^{P} is given in [88]. Let π_{H_e} denote a steady state probability for channel state $\vec{H}_e$ in the PSP/SSP cognitive wireless network given as

$$\pi_{H_e} = \begin{cases} p_H^{\mathrm{P}} \pi_H / p_H^{\mathrm{S}} \pi_H, & \text{when } \vec{H}_e = \vec{H} \\ 1 - p_H^{\mathrm{P}} / 1 - p_H^{\mathrm{S}}, & \text{when } \vec{H}_e = 0. \end{cases}$$

Additional modification of the channel model includes the option in which the channels towards the APs are owned by the SSP/PSP cognitive network and a separate band is used for inter terminal communication like Bluetooth or mmWave connections. This possibility is justified by the assumption that the pair-wise distances between the terminals are much shorter than the distances between the terminals and the AP. We call this option InTeNet (Inter Technology Networking) referring to networking of two different system concepts (e.g., Bluetooth or mmWave and cognitive cellular networking). One should note that given π_{H_e}, π_{H_e}, $\vec{H}_e$ and $\vec{H}$, the derivation of channel model in this case is straightforward.

As in Chapter 3, our analysis here also includes what we refer to as "partial cognitive networks" where the network operator's overall resources include both cognitive and conventional (purchased) links [85]. The modification of $\vec{H}_e(n)$ for such a case is also straightforward.

4.2.2 Cooperative Strategies

Each terminal has information of their own to send and they might like to cooperate in order to send this information to the AP at the highest rate possible. Thus, opportunistic cooperative control decisions within the cloud are required in order to maximize the long-term average throughput of the network and to maintain acceptably low levels of unfinished work in all queues. By cooperative decisions we mean that the cloud

members may choose to either relay data in a non-selfish cooperative manner or to reassign resources. It is assumed that terminal i can cooperate only with one terminal at time and that all terminals are willing to help each other inside the cloud, in order to get help themselves, when needed.

For each feasible cooperative pair (i, j) $(i, j \in \mathscr{I}, i \neq j)$ in frame n, we define a parameter $m_{ij}(n)$:

$$m_{ij}(n) = \begin{cases} 1, & \text{if terminal } i \text{ cooperates with user } j \\ 0, & \text{otherwise.} \end{cases}$$

The value of $m_{ij}(n)$ stays fixed during a frame but can change from frame to frame. Since each terminal i can cooperate only with one terminal at time, $\sum_j m_{ij}(n) = 1$. Let $\mathbf{M}(n)$ represent the cooperative communication matrix in frame n, defined as

$$\mathbf{M}(n) = \begin{pmatrix} m_{12}(n) & m_{13}(n) & \dots & m_{1|\mathscr{I}|}(n) \\ m_{21}(n) & m_{23}(n) & \dots & m_{2|\mathscr{I}|}(n) \\ \vdots & \vdots & & \vdots \\ m_{|\mathscr{I}|1}(n) & m_{|\mathscr{I}|2}(n) & \dots & m_{|\mathscr{I}|(|\mathscr{I}|-1)}(n) \end{pmatrix}.$$

Figure 12 illustrates the set of cooperative control options $\mathscr{V}$ for each cooperative pair (i, j). Let $|\mathscr{M}| = \lfloor |\mathscr{I}|/2 \rfloor$ denote the number of cooperating pairs inside a cloud. Each frame n is divided into $|\mathscr{M}|$ subframes n_{ij}, and each subframe is divided into 3 time slots t. In each subframe n_{ij}, user i always transmits in time slot 1 and user j $(j \in \mathscr{I}, j \neq i)$ always transmits in slot 2. This constraint arises from the fact that a terminal cannot transmit and receive information on the same frequency at the same time. In the third slot, terminals can cooperate either by relaying or reassigning resources. When terminals cooperate by relaying, terminal j/i helps terminal i/j in slot 3 by forwarding all packets terminal i/j has transmitted in one of the previous slots, and, simultaneously, user i/j repeats its own message from the previous slot. If terminals reassign resources, terminal i transmits both in its own and third slot of a frame or terminal j transmits in the second and the third slot of a subframe, as illustrated in Fig. 12. We use $V^{ij}(n) \in \mathscr{V}$ to represent a cooperative control decision of a cooperative pair (i, j) in frame n. For notational simplicity, $V^{ij}(n) = \{0, 1, 2, 3\}$ with values indicated in Fig. 12. Let $\vec{V}(n)$ represent the vector of cooperative control decisions of all cooperative pairs (i, j) in frame n.

We denote the transmitted signal of terminal i in slot t by $x_i(t)$ and the signal relayed by terminal j originated from terminal i by $x_{ij}(t)$. In addition, $\hat{y}(t)$ and $\hat{y}_i(t)$ are used

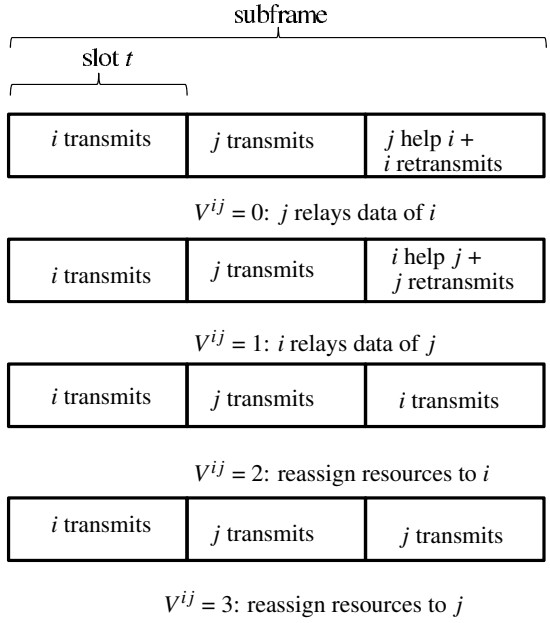

Fig. 12. Cooperative control options of a cooperative pair (i, j) in subframe n_{ij}.

to denote the received signal at AP and terminal i, respectively. Let w represent the complex circularly symmetric additive white Gaussian noise (AWGN) with zero mean and variance σ^2. If terminals cooperate by relaying, i.e., terminal j is chosen to help terminal i in the third slot ($i \neq j$), the input-output relationship for a subframe is given as

$$\hat{y}(t) = h_{i0}(t)x_i(t) + w_0(t) \tag{101}$$

$$\hat{y}_j(t) = h_{ij}(t)x_i(t) + w_j(t) \tag{102}$$

$$\hat{y}(t+1) = h_{j0}(t+1)x_j(t+1) + w_0(t+1) \tag{103}$$

$$\hat{y}_i(t+1) = h_{ji}(t+1)x_j(t+1) + w_i(t+1) \tag{104}$$

$$\begin{aligned}\hat{y}(t+2) &= h_{j0}(t+2)x_{ij}(t+2) + h_{i0}(t+2)x_i(t+2) + \\ w_0(t+2) &= h_{j0}(t+2)x_i(t+2) + h_{i0}(t+2)x_i(t+2) + w_0(t+2).\end{aligned} \tag{105}$$

We assume decode-and-forward transmission requiring both the relaying terminal and the AP to decode the entire codeword without errors.

If the third slot is assigned to terminal i so that terminal i transmits both in the first/second and the third slot of a subframe, the received signals are given as

$$\hat{y}(t) = h_{i0}(t)x_i(t) + w_0(t), \tag{106}$$

$$\hat{y}(t+1) = h_{j0}(t+1)x_j(t+1) + w_0(t+1) \tag{107}$$

and

$$\hat{y}(t+2) = h_{i0}(t+2)x_i(t+2) + w_0(t+2). \tag{108}$$

The strategies from Fig. 12 can be generalized by including the help to D users from $|\mathscr{I}|-1$ users by dividing each frame n into $F = |\mathscr{I}| + D$ time slots. The first $|\mathscr{I}|$ time slots are used for each user to transmit their own data. In the additional D slots, one slot is used to help a particular weak user in such a way that all users are repeating its signal. This is repeated in D time slots for D weakest users. If the system has only $d < D$ weak users whose signals in the first transmitted slots cannot provide necessary QoS, then the system needs to repeat only d transmissions with d possibly dynamically reconfigurable depending on the overall channel state, resulting in variable frame length $F = |\mathscr{I}| + d$. We call this strategy a reconfigurable cooperative strategy, which will be considered in our future work.

4.2.3 Channel Capacities

Let $B = 3|\mathscr{M}|$ denote the number of time slots t in frame n. In addition, we use $C_i^{V^{ij}}$ represent the channel capacity of user i for the given V^{ij}. Given the power constraint $P^{\text{tot}}(t)$ per user per slot and $\vec{H}(n)$, the channel capacities are given as

$$C_i^0 = \min\Big\{\frac{1}{B}\log_2\Big(1+\frac{|h_{ij}|^2 P^{\text{tot}}}{\sigma^2}\Big), \frac{1}{B}\log_2\Big(1+\frac{2|h_{i0}|^2 P^{\text{tot}} + |h_{j0}|^2 P^{\text{tot}}}{\sigma^2}\Big)\Big\}. \tag{109}$$

$$C_j^0(P^{\text{tot}}, H) = \frac{1}{B}\log_2\Big(1+\frac{|h_{j0}|^2 P^{\text{tot}}}{\sigma^2}\Big). \tag{110}$$

$$C_i^1(P^{\text{tot}}, H) = \frac{1}{B}\log_2\Big(1+\frac{|h_{i0}|^2 P^{\text{tot}}}{\sigma^2}\Big) \tag{111}$$

$$\begin{aligned} C_j^1(P^{\text{tot}}, H) = \min\Big\{&\frac{1}{B}\log_2\Big(1+\frac{|h_{ji}|^2 P^{\text{tot}}}{\sigma^2}\Big), \\ &\frac{1}{B}\log_2\Big(1+\frac{2|h_{j0}|^2 P^{\text{tot}} + |h_{i0}|^2 P^{\text{tot}}}{\sigma^2}\Big)\Big\}. \end{aligned} \tag{112}$$

$$C_i^2(P^{\text{tot}},H) = \frac{2}{B}\log_2\left(1+\frac{|h_{i0}|^2 P^{\text{tot}}}{\sigma^2}\right) \tag{113}$$

$$C_j^2(P^{\text{tot}},H) = \frac{1}{B}\log_2\left(1+\frac{|h_{j0}|^2 P^{\text{tot}}}{\sigma^2}\right) \tag{114}$$

$$C_i^3(P^{\text{tot}},H) = \frac{1}{B}\log_2\left(1+\frac{|h_{i0}|^2 P^{\text{tot}}}{\sigma^2}\right) \tag{115}$$

$$C_j^3(P^{\text{tot}},H) = \frac{2}{B}\log_2\left(1+\frac{|h_{j0}|^2 P^{\text{tot}}}{\sigma^2}\right) \tag{116}$$

Requiring the relaying users and the AP to decode the entire codeword without errors results in the minimum of the two capacity bounds in (109) and (112). We have assumed that the system bandwidth is 1 Hz.

4.2.4 Queuing Model

Assuming that $m_{ij}(n) = 1$, let $\mu_{i0}(n)$ denote the total service rate from terminal i to the AP, and $\mu_{ji}(n)$ is used to represent endogenous arrivals transmitted from terminal j to terminal i in frame n. Let $\vec{\mu}(n) = [\mu_{10}(n), \mu_{20}(n), ..., \mu_{|\mathscr{I}|0}(n)]$ represent the vector of service rates, and $\mu_i(n) = \mu_{i0}(n) - b_{ji}(n)\mu_{ji}(n)$, where $b_{ji}(n) = 1$, if user i relays the packets of terminal j in frame n, i.e., $V^{ij}(n) = 1$, or $b_{ji}(n) = 0$ otherwise ($V^{ij}(n) = 0$, $V^{ij}(n) = 2$ of $V^{ij}(n) = 3$). For simplicity, the service rates $\vec{\mu}(n)$ are restricted to integral multiples of packet lengths.

The arriving packets are placed into infinite transmission buffers that are assumed to be initially empty. Let $\vec{Q}(n) = [q_1(n), q_2(n), ..., q_{|\mathscr{I}|}(n)]$ represent the vector of queue lengths in frame n and $q_i(n)$ is used to denote the queue length of terminal i in frame n.

In the CWN, the queueing process is given as

$$q_i(n+1) = q_i(n) + a_i(n) - \mu_i(n). \tag{117}$$

In the PSP/SSP cognitive network, the queuing dynamics evolve as

$$q_i(n+1) = q_i(n) + a_i(n) - I(n)\mu_i(n), \tag{118}$$

where the channel corruption indicator is given by $I(n)$. If the transmission over the channel fails in the SSP/PSP cognitive network, $I(n) = 0$. In the PSP cognitive network

$I(n)$ is given as

$$I(n) = \begin{cases} 0; & \text{If SU returns to the channel and does not} \\ & \text{detect the presence of PU (collision)} \\ 1; & \text{Otherwise} \end{cases}$$

Let $p^{\mathrm{S}}_{\mathrm{return}}$ denote the probability that SU returns to the channel. Then, we have

$$p[I(n)=1] = (1-p^{\mathrm{S}}_{\mathrm{return}}) + p^{\mathrm{S}}_{\mathrm{return}} p_{\mathrm{sd}} \tag{119}$$

and

$$p[I(n)=0] = p^{\mathrm{S}}_{\mathrm{return}}(1-p_{\mathrm{sd}}), \tag{120}$$

where p_{sd} is the probability that SU correctly detects the presence of PU. For the given channel in the SSP cognitive network, the channel corruption indicator $I(n)$ is given as

$$I(n) = \begin{cases} 0; & \text{If PU returns to the channel} \\ 1; & \text{Otherwise} \end{cases}$$

where

$$p[I(n)=1] = 1 - p^{\mathrm{P}}_{\mathrm{return}} \tag{121}$$

and

$$p[I(n)=0] = p^{\mathrm{P}}_{\mathrm{return}}. \tag{122}$$

The PU and SU return probabilities $p^{\mathrm{P}}_{\mathrm{return}}$ and $p^{\mathrm{S}}_{\mathrm{return}}$ are discussed in [88].

Given the power constraint $P^{\mathrm{tot}}(t)$ per user per slot the queue length and $H(n)$, the service rates $\mu_i(n)$ should satisfy the following capacity constraint in frame n:

$$\mu_i(n) \leq C_i^{V^{ij}}(n). \tag{123}$$

It is also assumed that we cannot transmit more packets than there are in the queue.

4.3 Unified optimization problem

In order to derive a unified optimization problem for both CWN and PSP/SSP cognitive wireless networks, one should note that the service rate for PSP/SSP cognitive networks can be derived from the service rate of the CWN. When the service rate of terminal i in the CWN is given by $\mu_{i0}(n)$, the service rates in the PSP and PSP cognitive networks can be given as

$$\mu_{i0}(n)^* = \mu_{i0}(n) p^{\mathrm{P}}_H p[I(n)=1] \tag{124}$$

and

$$\mu_{i0}(n)^* = \mu_{i0}(n) p_H^{\mathrm{S}} p[I(n)=1], \tag{125}$$

respectively. In addition, let $\vec{\mu}(n)^* = [\mu_{10}(n)^*, ..., \mu_{|\mathscr{I}|0}(n)^*]$ to denote the vector of these service rates.

Let $y_i(n) = q_i(n) + a_i(n)$ and $\vec{Y}(n) = [y_1(n), ..., y_{|\mathscr{I}|}(n)]$ denotes the vector of $y_i(n)$s. We use $X(n) = \{\vec{Y}(n), \vec{H}(n)\}$ to denote the state of the system in frame n with countable state space $\mathscr{X}$. We use $\vec{V}_X(n)$ to represent the vector of control decisions of the $|\mathscr{M}|$ cooperative user pairs in frame n, when the state of the system is $X(n)$ and $\mathbf{M}_X(n)$ denote the cooperative communication matrix in state $X(n)$. At the beginning of each frame n, the network controller decides on the values of $\mathbf{M}_X(n)$ and $\vec{V}_X(n)$, and determines the transmission rates $\mu_{i0}(n)^*$ on each link by allocating a power vector $\vec{P}(n) = [P_1(n), P_2(n), ..., P_{|\mathscr{I}|}(n)]$ depending on the entire history of state evolutions. In addition, let $U_X(n) = \{\vec{\mu}(n)^*, \mathbf{M}_X(n), \vec{V}_X(n)\}$ represent a control input, i.e., an action, during frame n in state $X(n)$. The action $U_X(n)$ takes values in a general state space $\mathscr{U}_X$, which represents all feasible control options available under state $X(n)$. By feasible options we mean the set of control actions that satisfy the power and queue constraints, as we cannot transmit more packets than there are in the queue. Starting from state X, let $\pi = \{U_X(0), U_X(1), ...\}$ denote a policy, i.e., sequence of actions, that in frame $n, (n = 0, 1, ...)$, generates an action $U_X(n)$ depending on the entire history of previously chosen state-action pairs $U_X(\eta)$ for $\eta = \{0, 1, 2, ..., n-1\}$. Let Π denote the state space of all such policies. We assume that centralized control is possible so that the network controller has access to full backlog and channel state information.

For policy π, let us now define a parameter z_i as

$$z_i = \frac{\lambda_i}{\lim_{n\to\infty} \frac{1}{n} \sum_{\eta=0}^{n-1} \mu_i(\eta)^*}. \tag{126}$$

The control problem is to map from the current queue states and channel gains to an optimal sequence of $U_X(n)$, i.e., policy π, that stabilizes the system and solves the following optimization problem:

$$\begin{aligned} \underset{\pi\in\Pi}{\text{maximize}} \quad & \lim_{n\to\infty} \frac{1}{n} \sum_{\eta=0}^{n-1} \mathbb{E}_X^{\pi} \Big\{ \sum_{i\in|\mathscr{I}|} \mu_i(\eta)^* \Big\} \\ \text{subject to} \quad & \max\{z_1, ..., z_{|\mathscr{I}|}\} \leq 1 \end{aligned} \tag{127}$$

In (127), the control decisions are made based on queue length and channel state information of the terminals within the cloud. Thus, the terminals with a short/empty

queue and a good channel can help the terminals with a long queue and a bad channel by relaying packets or by letting the weakest terminals to use the additional time slot. The idea is to efficiently allocate system resources resulting in significant gains in several metrics, especially, when the cooperating terminals have different user demands.

4.4 Optimal Control Policy

The control problem given in (127) is a constrained dynamic optimization problem. One way to solve it is to convert it into an unconstrained problem [41], [42]. The unconstrained problem is a standard Markov Decision Process (MDP) and we define the optimal policy for this MDP using the Value iteration algorithm [15], [94].

4.4.1 Formulation as a Markov Decision Process

The set of feasible actions U_X in each state $X = \{\vec{Y}, \vec{H}\}$ is the set of all actions that satisfy the constraint in (123) and we cannot transmit more packets than there are in the queue. After taking an action U_X, the following state is denoted as S. We now let $p(S|X,U_X)$ to denote the transmission probability from state X to state S with action U_X.

For a policy $\pi \in \Pi$, we define reward and cost functions as

$$D = \lim_{n\to\infty} \frac{1}{n} \sum_{\eta=0}^{n-1} \mathbb{E}_X^{\pi}\{ \sum_{i\in|\mathscr{I}|} \mu_i(\eta)^*\} \tag{128}$$

and

$$E = \max\{z_1, ..., z_{|\mathscr{I}|}\}, \tag{129}$$

respectively. Given the constraints in (127), let Π_E denote the set of admissible control policies $\pi \in \Pi$ that satisfy (109) - (116) and $E \leq 1$. Then, the objective can be restated as a constrained optimization problem given as [15]

$$\text{maximize } \; D; \;\; \text{subject to } \; \pi \in \Pi_E. \tag{130}$$

The problem in (130) is converted into a family of unconstrained optimization problems through a Lagrangian relaxation [93]. Given (117) and (118), it is easy to see that a policy that minimizes $\max\{z_1, ..., z_{|\mathscr{I}|}\}$ is equivalent to a policy that is designed to minimize $\max\{q_1(n+1)\}$, ..., $q_{|\mathscr{I}|}(n+1)\}$ in every frame. Thus, the corresponding

Lagrangian function for any policy $\pi \in \Pi$ and $\beta \geq 0$ can now be defined as,

$$J_{\beta}^{\pi}(X) = \lim_{n\to\infty} \frac{1}{n} \sum_{\eta=0}^{n-1} \mathbb{E}_{X}^{\pi}\Big\{ \sum_{i\in|\mathscr{I}|} \mu_i(\eta)^* - \beta \max\{q_1(\eta+1), ..., q_{|\mathscr{I}|}(\eta+1)\}\Big\}. \quad (131)$$

The Lagrangian multiplier β indicates the relative importance of queue lengths over the throughput; larger value of β corresponds to placing more importance on keeping the queue lengths short. Given $\beta \geq 0$, we define the unconstrained optimization problem as

$$\text{maximize} \;\; J_{\beta}^{\pi}(X) \;\; \text{subject to} \;\; \pi \in \Pi. \quad (132)$$

An optimal policy for unconstrained problem is optimal also for the original constrained control problem when β is appropriately chosen [15], [93].

The problem given in (132) is a standard MDP with maximum average reward criterion. For each initial state X, we define the corresponding discounted cost MDP with value function

$$W_{\alpha}(X) = \underset{\pi\in\Pi}{\text{maximize}} \;\; \mathbb{E}_{X}^{\pi}\left\{ \sum_{\eta=0}^{\infty} \alpha^{\eta} R[X(\eta), U_X(\eta)] \right\}, \quad (133)$$

where the discount factor $\alpha \in (0,1)$, and the reward from taking an action U_X in state X is defined as

$$R[X(\eta), U_X(\eta)] = \sum_{i\in|\mathscr{I}|} \mu_i(\eta)^* - \beta \max\{q_1(\eta+1), ..., q_{|\mathscr{I}|}(\eta+1)\}. \quad (134)$$

The value function $W_{\alpha}(X)$ can be defined as the optimal total expected discounted reward for discount factor α [94]. One way to solve (133) is to use value iteration algorithm (VIA) [15], [94].

VIA is a standard dynamic programming approach to recursively compute an ε-optimal policy π^* for (133) [95]. For notational simplicity, we drop the subscript α. The solution to (133), i.e., the optimal value functions $W^*(X)$ for each initial state X and the corresponding discount optimal policies $\pi^* \in \Pi$, can be solved with the following iterative algorithm:

$$W^{l+1}(X) = \max_{U_X\in\mathscr{U}_X} \Big\{ R(X,U_X) + \alpha \sum_{S\in\mathscr{X}_S} p(S|X,U_X) W^{l}(S) \Big\}, \quad (135)$$

where $\mathscr{X}_S$ is the set of feasible states that follow state X by taking an action U_X, and $p(S|X,U_X)$ is the transition probability from state X to state S with action U_X. For each initial state X, define the best action U_X for each state X as

$$\arg\max_{U_X\in\mathscr{U}_X} \left\{ R(X,U_X) + \alpha \sum_{S\in\mathscr{X}_S} p(S|X,U_X) W^{*}(S) \right\}. \quad (136)$$

4.5 Achievable rates

In this section, we characterize the fundamental throughput limitations and establish the unified stability regions of cooperative and NC communication systems for both SSP, PSP cognitive networks, and for CWN. In addition, the concept of InSyNet is used to derive an upper bound for the stability region of the cooperative system with $|\mathscr{I}|$ users inside the cloud. As a performance measure we use a parameter referred to as "harvested capacity", which is defined as difference in the stability region of the new cooperative strategies and the stability region of the conventional NC system.

4.5.1 A unified cooperative network stability region

Given the power constraint P^{tot} per user per slot, let $\mathscr{U}^H_{P^{\text{tot}}}$ denote the set of all possible resource allocation options in channel state $\vec{H}$ that satisfy the power constraint P^{tot}, and $U^H_{P^{\text{tot}}} \in \mathscr{U}^H_{P^{\text{tot}}}$ represents a control action in channel state $\vec{H}$. Given the power constraint P^{tot} per user per slot, let g^*_{i0} denote the long-term average rate that can be supported by the channel between terminal i and the AP and g^*_{ji} denotes the long-term average rate that is supported by the channel between terminals j and i. Let $\vec{G}^* = [g^*_1, g^*_2, ..., g^*_{|\mathscr{I}|}]$ denote the vector of average long-term supportable data rates, where $g^*_i = g^*_{i0} - g^*_{ji}$. We use g_i to represent the long-term rate in the CWN. For PSP and SSP cognitive networks, g^*_i is given as

$$g^*_i = g_i p^{\text{P}}_H p(I = 1) \tag{137}$$

and

$$g^*_i = g_i p^{\text{S}}_H p(I = 1), \tag{138}$$

respectively.

Due to the time varying system state conditions, $\vec{G}^*$ must be averaged over all possible channel states. Moreover, $\vec{G}^*$ is not fixed and depends on transmission policy for choosing the best actions for each $\vec{H} \in \mathscr{H}$. Since the numerical calculation of all supportable rates $\vec{G}^*$ is computationally very challenging, we simplify the problem and consider only the policies that use the maximum power P^{tot} in every time slot. Assuming orthogonal scheduling of the cooperative users, the network stability region can be defined by considering only the set of policies where each active user i uses maximum transmit power in slot t. Let $\widehat{\mathscr{U}}^H_{P^{\text{tot}}} \subset \mathscr{U}^H_{P^{\text{tot}}}$ represent the set of all possible control actions in channel state $\vec{H}$ that use maximum available power in every time slot, and $\widehat{U}^H_{P^{\text{tot}}} \in \widehat{\mathscr{U}}^H_{P^{\text{tot}}}$ denote a specific control action belonging to the set $\widehat{\mathscr{U}}^H_{P^{\text{tot}}}$. In addition,

let $\widehat{G}^* = [\widehat{g}_1^*, ..., \widehat{g}_{|\mathscr{I}|}^*]$ represent the long-term average transmission rate vector for the full power policies. The set of all full power long-term average transmission rates $\widehat{G}^*$ that the network can be configured to support, can now be written as:

$$\Gamma^* = \sum_{H \in \mathscr{H}} \pi_H \text{Conv}\{\mu(\widehat{U}_{P\text{tot}}^H, H)^* | \widehat{U}_{P\text{tot}}^H \in \widehat{\mathscr{U}}_{P\text{tot}}^H\}, \tag{139}$$

where the addition and scalar multiplication of sets is used, $\text{Conv}\{\mathscr{B}_H\}$ represents convex hull of the set $\mathscr{B}_H$ that is defined as the set of all convex combinations $p_1 b_1 + p_2 b_2 + ... + p_j b_j$ of elements $b_j \in \mathscr{B}_H$ and p_js are probabilities summing to 1. Specifically, Γ^* represents the set of data rates for set of policies that use maximum power in every time slot and it is a subset of the total supportable arrival rate region.

The throughput region in (139) can be seen as the set of all long-term full power average service rates. Thus, the unified stability region of the cooperative communication system Λ_{C} is the set of all arrival rates vectors $\vec{\lambda} = [\lambda_1, ..., \lambda_{|\mathscr{I}|}]$ for which there exists a transmission policy π that satisfies

$$\lambda_i \leq \lim_{n\to\infty} \frac{1}{n} \sum_{\eta=1}^{n-1} \mathbb{E}_X^{\pi}\{\mu_i(\eta)^*\} \leq \sum_H \pi_H \widehat{g}_H^* \tag{140}$$

for some $\widehat{G}^* = \sum_H \pi_H \widehat{G}_H^* \in \Gamma^*$, as rates below each point in Γ^* can likewise be supported. The average long-term supportable service rates in channel state $\vec{H}$ is given by $\widehat{G}_H^*$. Specifically, $\vec{\lambda}$ is in the region Λ_{C} if there exists a long-term average rate vector $\vec{G}^*$ such that there exists a transmission process which supports the rates $\vec{\lambda}$.

In order to facilitate the illustration of the unified stability region, we fix $|\mathscr{I}| = 2$. Let λ_i^{p}, λ_i^{s} denote the supportable rates in PSP and SSP cognitive networks, respectively. For the channel model given in Section 4.7, the unified cooperative stability region Λ_{C} is illustrated in Fig. 13, where λ_i is given as

$$\lambda_i = \lambda_i^{\text{p}} / p_H^{\text{P}} p(I = 1) \tag{141}$$

for PSP cognitive network and

$$\lambda_i = \lambda_i^{\text{s}} / p_H^{\text{S}} p(I = 1) \tag{142}$$

for SSP cognitive network. One should not that $\vec{\lambda} \in \Lambda_{\text{C}}$ is a necessary condition for stability and $\vec{\lambda}$ strictly interior to Λ_{C} is a sufficient condition for the system to be stabilized by a transmission policy [12].

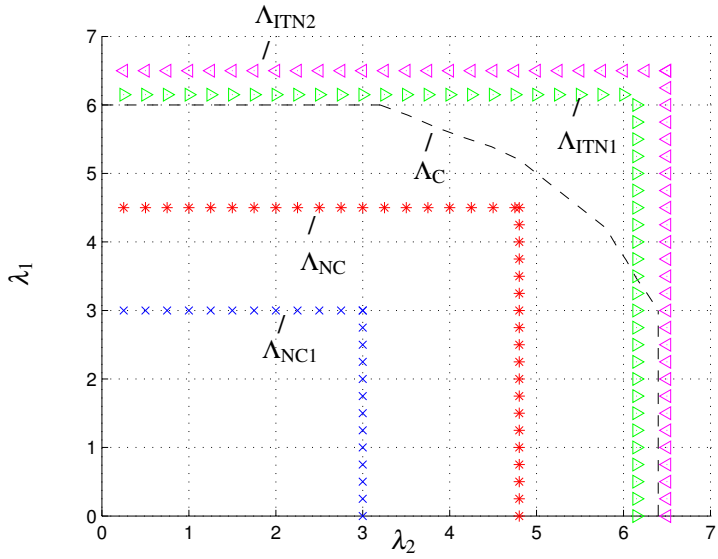

Fig. 13. Unified network stability regions.

4.5.2 InTeNet:Inter Technology Networking

For comparison, InTeNet concept that is a hybrid of mmWave and macrocell technology is introduced to derive an upper bound for the capacity region of the proposed cooperative communication network. Within a mobile cloud of $|\mathscr{I}|$ users, mmWave or Bluetooth connections are used in such a way that the users mutually exchange their packets to be transmitted to the AP of the macro network. The incoming mmWave technology can provide high capacity on short distances and can be used to exchange the data between the users within the cloud. We can implement this by assuming that the channel between the users within the cloud is ideal so that infinitesimal slot (of negligible length) is needed to exchange the data among the users within the cloud. Then, the user with the best macro channel transmits all data from the cloud to the AP. If the Bluetooth or separate mmWave channel is used, then intercloud and cloud to AP transmissions can be active simultaneously.

We start with establishing the stability region of InTeNet, when $|\mathscr{I}| = 2$ and both users are assumed to have infinite buffers. Users within the cloud know each other's data and the maximum supportable rate is achieved when a user with the best channel transmits the data of both users to the AP. Thus, the maximum supportable long-term

average arrival rate for the 2-user InTeNet is given as

$$\lambda_i \leq \frac{1}{2} \sum_{H \in \mathscr{H}} \pi_H \max\{\log_2(1+|h_{10}|^2 P^{\text{tot}}), \log_2(1+|h_{20}|^2 P^{\text{tot}})\}. \tag{143}$$

For both PSP and SSP cognitive networks as well as CWN, the unified stability region of the two-user InTeNet, Λ_{ITN1}, is illustrated in Fig. 13, where λ_i is given in (141) and (142) for PSP and SSP cognitive networks, respectively.

As $|\mathscr{I}|$ grows, the probability that one of the users within the cloud has the best possible channel increases, and the maximum supportable rate is given as

$$\begin{aligned}\lambda_i \leq \frac{1}{|\mathscr{I}|} \sum_{H \in \mathscr{H}} \pi_H \max \Big\{ & \log_2(1+|h_{10}|^2 P^{\text{tot}}), \\ & \log_2(1+|h_{20}|^2 P^{\text{tot}}), ..., \log_2(1+|h_{|\mathscr{I}|0}|^2 P^{\text{tot}}) \Big\}.\end{aligned} \tag{144}$$

The upper bound for the stability region of the cooperative communication network can be derived by considering the stability region of an InTeNet, where $|\mathscr{I}| \to \infty$. In this case, each user within the cloud can use the best possible channel between the users and the AP in each frame given as $h^{\max} = \max_{\{i, H \in \mathscr{H}\}}\{h_{i0}\}$. The overall capacity from the cloud to the AP in frame n is then

$$\sum_{i \in \mathscr{I}} \frac{1}{|\mathscr{I}|} \log_2(1+|h^{\max}|^2 P^{\text{tot}}) = \log_2(1+|h^{\max}|^2 P^{\text{tot}}) \tag{145}$$

as opposed to the approach without Bloethoot or mmWave, where

$$\sum_{i \in \mathscr{I}} \frac{1}{|\mathscr{I}|} \log_2(1+|h_{i0}|^2 P^{\text{tot}}). \tag{146}$$

Let Λ_{ITN2} represent the stability region of InTeNet, when $|\mathscr{I}| \to \infty$. For fair comparison, we assume that there are only two users within the infinite user cloud that have external arrivals. The maximum supportable arrival rate region Λ_{ISN2} can now be illustrated in Fig. 13, that is given as

$$\lambda_i \leq \frac{1}{2}\{\log_2(1+|h^{\max}|^2 P^{\text{tot}})\}. \tag{147}$$

4.5.3 Unified non-cooperative network stability region

Let us now consider the NC communication system with queuing, where only direct communication with the AP is allowed. We use $g_{\text{NC}i}^{H^*}$ to denote the maximum average

supportable rate with power P^{tot} under channel state $\vec{H}$, where the subindex NCi stands for non-cooperation of user i. For the PSP cognitive network, $g^{H^*}_{\text{NC}i}$ is given as

$$g^{H^*}_{\text{NC}i} = g^{H}_{\text{NC}i} p_1 p^{\text{P}}_H p(I=1), \tag{148}$$

and for the SSP cognitive network

$$g^{H^*}_{\text{NC}i} = g^{H}_{\text{NC}i} p^{\text{S}}_H p(I=1). \tag{149}$$

The average maximum supportable long-term rate of a two-user non-cooperative network is represented as

$$\sum_H \pi_H \frac{1}{2}(C_i^2(P^{\text{tot}},H)^* + C_i^3(P^{\text{tot}},H)^*) = \sum_H \pi_H g^{H^*}_{\text{NC}i}, \tag{150}$$

where $C_i^2(P^{\text{tot}},H)^* = p^{\text{P}}_H p(I=1) C_i^2(P^{\text{tot}},H)$ and $C_i^3(P^{\text{tot}},H)^* = p^{\text{S}}_H p(I=1) C_i^2(P^{\text{tot}},H)$ for the PSP and SSP cognitive networks, respectively. In order for the non-cooperative system to be stable, λ_i must satisfy

$$\lambda_i \leq \lim_{n\to\infty} \frac{1}{n} \sum_{\eta=0}^{n-1} \mathbb{E}^{\pi}_X \{\mu_i(\eta)^*\} \leq \sum_H \pi_H g^{H^*}_{\text{NC}i}. \tag{151}$$

For the PSP and SSP cognitive networks as well as for the CWN, the unified stability region of the proposed two-user non-cooperative network, Λ_{NC}, is plotted in Fig. 13, where it can be compared to the corresponding stability region of the cooperative communication network Λ_{C}. The difference between Λ_{C} and Λ_{NC} is referred to as "harvested capacity". It can be seen from the figure that the capacity region achieved by the cooperative strategy is approximately 38% larger than the capacity region of the non-cooperative network.

Let us now consider a two-user delay-limited non-cooperative communication system where neither dropping the packets nor queuing is allowed. Let Λ_{NC1} denote the stability region of the delay-limited non-cooperative system. Intuitively, the maximum supportable rate of the delay-limited system is

$$\lambda_i \leq \frac{1}{2} \min_{\{i,H\in\mathscr{H}\}} \log_2(1+|h_{i0}|^2 P^{\text{tot}}), \tag{152}$$

that is also illustrated in Fig. 13.

4.6 Unified stability analysis

In this section, a unified stability analysis for both PSP and SSP cognitive networks as well as for CWNs is presented.

We show that if the control actions need to be calculated for each state X like in Section 4.4, the best network stabilizing policy minimizes the maximum queue length over all the terminals. Finally, we show that our dynamic control policy stabilizes the network.

4.6.1 The best network stabilizing policy

The best network stabilizing policy is usually considered to be the one that minimizes the average queue length for each terminal. However, in our case, where the control actions need to be calculated for each state X, the best network stabilizing policy is the one that minimizes the maximum queue length over all terminals.

When the optimal control actions need to be calculated for each state X, like in Section 4.4, we can minimize the complexity by minimizing the number of states for which we need to calculate the action. Thus, the smaller the maximum queue length, the better. Thus, the best network stabilizing policy minimizes

$$q^{\max} = \max\{q_1^{\max}, q_2^{\max}, ..., q_{|\mathscr{I}|}^{\max}\}, \tag{153}$$

where

$$q_i^{\max} = \limsup_{n\to\infty} q_i(n). \tag{154}$$

Let $n_i^{\max}$ represent the index of the frame, when $q_i(n) = q_i^{\max}$. The maximum queue length can be now given as

$$q_i^{\max} = q_i(n_i^{\max}) = q_i(0) + \sum_{\eta=0}^{n_i^{\max}-1} a_i(\eta) - \mu_i(\eta)^*. \tag{155}$$

Now, inserting (155) into the right hand side of (153), (153) can be rewritten as

$$\max\{q_1(0) + \sum_{\eta=0}^{n_1^{\max}-1} a_1(n) - \mu_1(n)^*, ..., q_{|\mathscr{I}|}(0) + \sum_{\eta=0}^{n_{|\mathscr{I}|}^{\max}-1} a_{|\mathscr{I}|}(n) - \mu_{|\mathscr{I}|}(n)^*\}. \tag{156}$$

The whole network can be defined to be stable, when $q^{\max} < \infty$. Thus, when analysing the network stability, it is sufficient to consider only the stability of $q^{\max}$.

4.6.2 The K-step Lyapunov drift for q^{max}

The maximum transmission rate out of a node is $\mu_{\max}^{\text{out}} \triangleq \max_{\{i\in\mathscr{I},H\in\mathscr{H},U_{P^{\text{tot}}}^{H}\in\mathscr{U}_{P^{\text{tot}}}^{H}\}} \mu_i(U_{P^{\text{tot}}}^{H},H)$. Such a value exists because $\mu_i(U_{P^{\text{tot}}}^{H},H)$ is bounded [14], [12]. Based on $\vec{H}_e$ given in Subsubection 4.2.1, $\mu_{\max}^{\text{out}}$ is the same both for SSP/PSP cognitive networks as well as for CWN.

Consider the unified K-step dynamics of $q^{\max}$ for both SSP/PSP cognitive networks and for CWNs:

$$q^{\max}(n_0+K) = q^{\max}(n_0) + \sum_{n=n_0}^{n_0+K-1} \max_{i\in\mathscr{I}}\{a_i(n)-\mu_i(n)^*\}. \tag{157}$$

Squaring both sides of (157), defining the Lyapunov function as $L(q^{\max}) = (q^{\max})^2$ and taking the conditional expectation given $q^{\max}(n_0)$, the K-step Lyapunov drift can now be given as

$$\mathbb{E}\{L[q^{\max}(n_0+K)] - L[q^{\max}(n_0)]|q^{\max}(n_0)\} \leq K^2M - 2q^{\max}(n_0)\mathbb{E}\{\sum_{n=n_0}^{n_0+K-1} \max_{i\in\mathscr{I}}\{\mu_i(n)^* - a_i(n)\}|q^{\max}(n_0)\}, \tag{158}$$

where

$$M \triangleq (a_{\max})^2 + (\mu_{\max}^{\text{out}})^2, \tag{159}$$

$a_{\max} < \infty$ and $n_0 \in \{0,1,...,K-1\}$.

The inequality in (158) represents the K-step Lyapunov drift for any resource allocation policy yielding transmission rate vector $\vec{\mu}(n)^*$ and it was first presented in [20]. The intuition behind the drift is that when the queue length gets sufficiently large, the right hand side of (158) gets negative, leading to negative feedback and stability [12], [20].

4.6.3 Network stabilizing policy

In this subsection, we analyse the stability of the proposed dynamic control policy.

Theorem 3. *Our dynamic transmission policy minimizes (153) and stabilizes the network.*

Proof. Specifically, our dynamic control policy π^* is designed to maximize

$$\frac{1}{K}\sum_{n=n_0}^{n_0+K-1}\sum_{i}\mathbb{E}\big\{\mu_i(n)^* - \beta\max\{q_1(n+1),...,q_{|\mathscr{I}|}(n+1)\}\big\}. \tag{160}$$

Inserting $q_i(n+1) = q_i(n_0) + \sum_{\eta=n_0}^{n}[a_i(\eta) - \mu_i(\eta)^*]$ into (160), we get

$$\frac{1}{K}\sum_{n=n_0}^{n_0+K-1}\sum_{i}\mathbb{E}\Big\{\mu_i(n)^* - \beta\max\Big\{q_1(n_0) + \sum_{\eta=n_0}^{n}[a_1(\eta) - \mu_1(\eta)^*], ..., q_{|\mathscr{I}|}(n_0) + \sum_{\eta=n_0}^{n}[a_{|\mathscr{I}|}(\eta) - \mu_{|\mathscr{I}|}(\eta)^*]\Big\}\Big\}. \quad (161)$$

It is easy to see that our dynamic policy is designed to minimize (156). It is also easy to see that since the dynamic policy allocates the resources so that $\sum_{\eta=n_0}^{n}[a_i(\eta) - \mu_i(\eta)^*]$ is minimized over all terminals, the dynamic policy is designed to maximize the right hand side of (158). Thus, the long-term average maximum queue length is minimized and, if the arrival rates are inside the network stability region, our dynamic policy provides a stable network. □

4.7 Performance evaluation

For illustration purposes and to validate our stability analysis in Sections 4.5 and 4.6, we evaluate the performance of the optimal dynamic policy π^* with simulations. The resulting power, delay and throughput curves of the cooperative policy in the CWN are compared with the corresponding NC case, where only direct transmission control option is allowed. Our cooperative policy is shown to achieve significant power savings and delay improvements over the corresponding NC network. In addition, the performance of the optimal cooperative policy is evaluated in the presence of uncertain link availability and reliability both in cognitive and partial cognitive (PC) networks. It is shown that by adapting to the changes in network conditions, our cooperative control policy mitigates the effect of PSP and SSP CNs on each other. For simplicity, $|\mathscr{I}| = 2$.

The channel process is generated according to a Markov chain and the channel state transition matrix for the CWN is

$$\mathbf{T} = \begin{bmatrix} T_{11} & T_{12} & T_{13} & T_{14} \\ T_{21} & T_{22} & T_{23} & T_{24} \\ T_{31} & T_{32} & T_{33} & T_{34} \\ T_{41} & T_{42} & T_{43} & T_{44} \end{bmatrix} = \begin{bmatrix} 0.3 & 0.5 & 0.2 & 0 \\ 0.1 & 0.6 & 0.2 & 0.1 \\ 0.1 & 0.3 & 0.5 & 0.1 \\ 0 & 0.1 & 0.25 & 0.65 \end{bmatrix}, \quad (162)$$

where $T_{\dot{h},\ddot{h}}$ is the probability of transition from channel state $\dot{h}$ to $\ddot{h}$, and the corresponding stationary probabilities $p\{\vec{H} = [|h_{10}|^2, |h_{20}|^2, |h_{12}|^2, |h_{21}|^2]\}$ are given as $p\{\vec{H} = [10, 10, 500, 500]\} = 0.1$, $p\{\vec{H} = [10, 100, 500, 500]\} = 0.4$, $p\{\vec{H} = [100, 10, 500, 500]\} = 0.3$ and $p\{\vec{H} = [100, 100, 500, 500]\} = 0.2$.

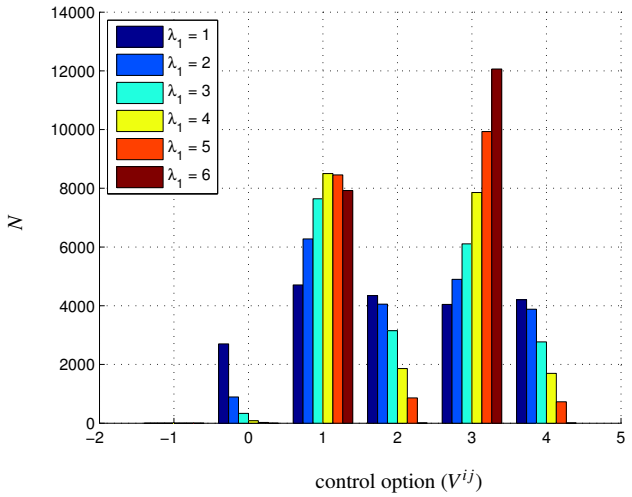

Fig. 14. Histogram of the cooperative control decisions of the dynamic control algorithm in the CWN, when $\lambda_2 = 1$.

If the channel is used within the SSP cognitive network, p_H^{S} in (99) equals 0.9 or 0.7, and $p\{\vec{H} = [10, 10, 500, 500]\} = 0.09$, $p\{\vec{H} = [10, 100, 500, 500]\} = 0.36$, $p\{\vec{H} = [100, 10, 500, 500]\} = 0.27$, $p\{\vec{H} = [100, 100, 500, 500]\} = 0.18$, $p\{\vec{H} = [0, 0, 500, 500]\} = 0.1$ or $p\{\vec{H} = [10, 10, 500, 500]\} = 0.07$, $p\{\vec{H} = [10, 100, 500, 500]\} = 0.28$, $p\{\vec{H} = [100, 10, 500, 500]\} = 0.21$, $p\{\vec{H} = [100, 100, 500, 500]\} = 0.14$, $p\{\vec{H} = [0, 0, 500, 500]\} = 0.3$. For the given channel, the probability that $I = 1$ in (121) is $p(I = 1) = 0.95$.

In the PC network, where the overall resources include both cognitive and conventional links, we assume that the channel between the U2 and the AP is cognitive and the channel between U1 and the AP is non-cognitive. The probability that the channel between U2 and AP is available for communication is 0.9 or 0.7, and $p\{\vec{H} = [10, 10, 500, 500)\} = 0.09$, $p\{\vec{H} = [10, 100, 500, 500)\} = 0.36$, $p\{\vec{H} = [100, 10, 500, 500)\} = 0.27$, $p\{\vec{H} = [100, 100, 500, 500]\} = 0.18$, $p\{\vec{H} = [100, 0, 500, 500]\} = 0.05$, $p\{\vec{H} = [10, 0, 500, 500]\} = 0.05$ or $p\{\vec{H} = [10, 10, 500, 500]\} = 0.07$, $p\{\vec{H} = [10, 100, 500, 500]\} = 0.28$, $p\{\vec{H} = [100, 10, 500, 500]\} = 0.21$, $p\{\vec{H} = [100, 100, 500, 500]\} = 0.14$, $p\{\vec{H} = [100, 0, 500, 500]\} = 0.15$, $p\{\vec{H} = [10, 0, 500, 500]\} = 0.15$. For the given channel between U2 and AP, the probability that $I = 1$ in (**??**) is $p(I = 1) = 0.95$.

For a Poisson distributed process, the second moment of arrivals in each frame is finite [12]. Thus, the arrivals are bounded in their second moments every time slot and

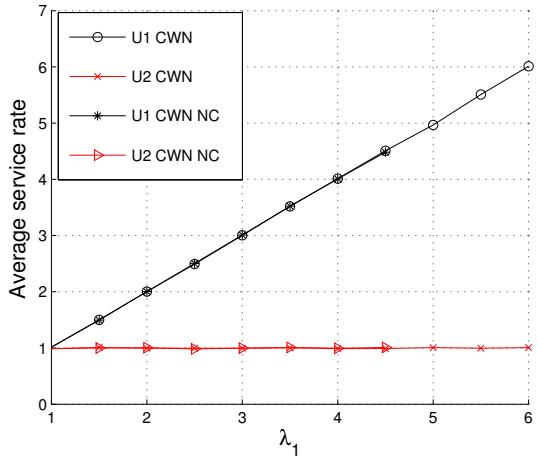

Fig. 15. Average service rates for the optimal policy in the CWN as a function of λ_1 for both cooperative and NC systems, when $\lambda_2 = 1$.

$\mathbb{E}\{[a_i(n)]^2\} \leq a_{max}^2$ according to a Poisson distributed process. The average arrival rate of user 2 (U2) is fixed to 1 packet/frame. The average arrival rate of user 1 (U1), λ_1, gets values inside the network stability region, i.e., λ_1 varies between 1 to 6 packets/frame. The discount factor in (133) is α =0.7 and the Lagrangian multiplier in (132) is defined as $\beta = [0.5, 0.5]$. The long-term average power, delay, and throughput are calculated over $N = 20000$ frames.

Fig. 14 illustrates the histogram of the cooperative control decisions of the optimal dynamic policy in the CWN, when λ_1 varies from 1 to 6. On the horizontal axis, the control options are represented as: 0 corresponds to a case where no packets are transmitted, 1 corresponds to $V^{ij} = 0$ (U2 helps U1 in slot 3), 2 corresponds to $V^{ij} = 1$ (U1 helps U2 in slot 3), 3 corresponds to $V^{ij} = 2$ (U1 transmits in both slot 1 and slot 3), and 4 corresponds to $V^{ij} = 3$ (U2 transmits both in slot 2 and slot 3). It can be seen in the figure, that the number of $V^{ij} = 0$ and $V^{ij} = 2$ actions increases, and the number of $V^{ij} = 1$ and $V^{ij} = 3$ actions decreases, as λ_1 increases. This is because U2 starts to help U1 with the increase of λ_1 since it is more beneficial (in terms of stability and throughput) for the system to choose more $V^{ij} = 0$ and $V^{ij} = 2$ actions as the queue length of U1 grows.

In Fig. 15, the long-term average service rates of the optimal control algorithm in the CWN for both cooperative and NC systems are plotted as a function of λ_1, when λ_2 was fixed to 1 packet/frame. It can be seen from the figure that the average service rate

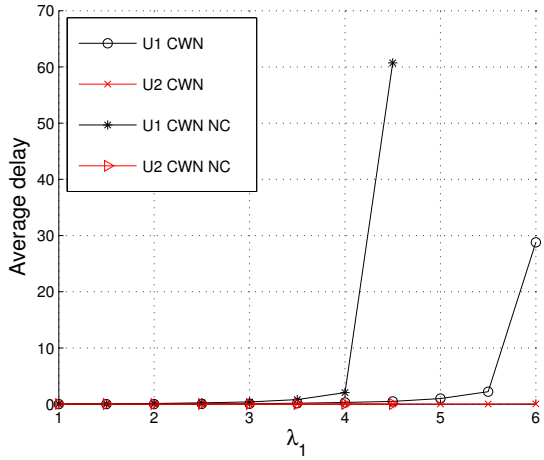

Fig. 16. Average delay for the optimal policy in the CWN as a function of λ_1 for both cooperative and NC systems, when $\lambda_2 = 1$.

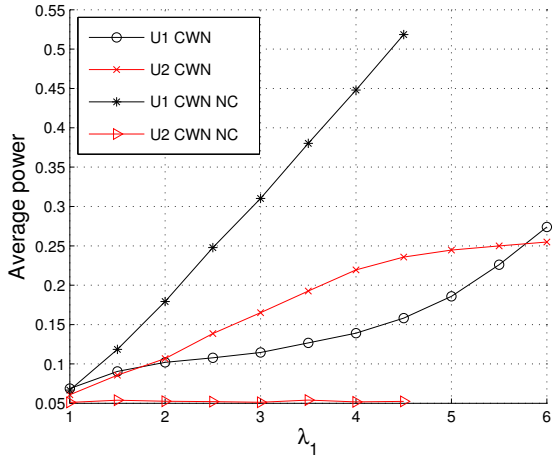

Fig. 17. Average powers for the optimal policy in the CWN as a function of λ_1 for both cooperative and NC systems, when $\lambda_2 = 1$.

equals λ_i for both cooperative and NC systems. This supports the stability analysis in Section 5.7, since $\lambda_1 = 6$ and $\lambda_1 = 4.5$ are on the boundaries of the stability regions of the cooperative and NC networks illustrated in Fig. 13.

The long-term average delays of the optimal control policy in the CWN for both cooperative and NC systems are plotted as a function of λ_1 in Fig. 30. The arrival rate

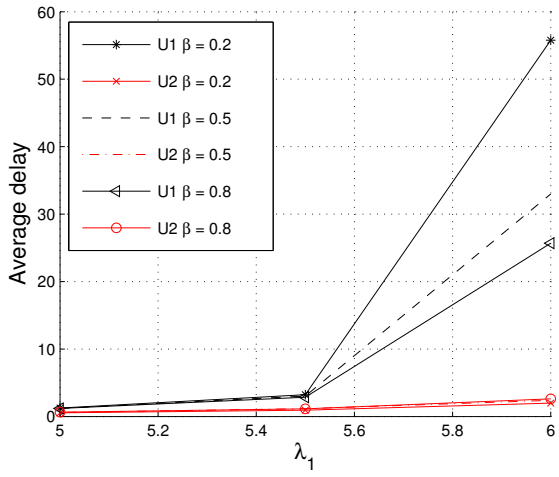

Fig. 18. Average delays of the optimal cooperative policy in the CWN as a function of λ_1 for different values of β, when $\lambda_2 = 3$.

of U2 was fixed to 1 packet/frame. Due to the low λ_2, the curves (U2 CWN NC) and (U2 CWN) have values close to zero. It can also be seen that the help U2 provides to U1 does not have affect on the delay of U2. However, it can be seen that cooperative communication provides significant reduction in the delay of U1 compared with the NC system. Due to the very different user demands, U2 can help U1 in the cooperative communication network so that the delay of U2 stays low. In addition, the results in Fig. 30 support the stability analysis in Section 5.7, since the average delays in Fig. 30 are bounded on the boundaries of the stability regions of the cooperative communication and NC communication networks.

Fig. 31 illustrates the long-term average powers of the dynamic control algorithm in the CWN versus λ_1, when $\lambda_2 = 1$. It can be seen that the power of U1 required for cooperative communication is significantly smaller than the power of U1 used for NC communication. However, due to the help U2 provides to U1 in the cooperative communication system, U2 requires more power in the cooperative case than in the NC case. The difference between these two curves represents the power that U2 use for helping U1 in the cooperative network. Nevertheless, one should notice that the increase in the power of U2 in the cooperative communication network is still lower than the power savings of U1 which results in overall power savings in the cooperative communication system. Without any loss in throughput or the delay of U2 in the

cooperative communication network, U2 can utilize some of its power to cooperate with U1, since λ_2 is low. As mentioned before, it becomes more likely to choose action $V_X = 2$ than action $V_X = 0$ when λ_1 is high. That is why U1 CWN curve crosses U2 CWN curve as λ_1 gets high.

In Fig. 18, the long-term average delays of U1 and U2 are plotted as a function of λ_1 for different values of β in the CWN, when $\lambda_2 = 3$. Larger values of β correspond to placing more importance on queue lengths increasing the probability to choose an action that keep the delays of both users low. That is why the delay of U1 decreases as the value of β increases at high λ_1 values. The procedure to calculate the optimal β is omitted due to the page limit and computational complexity of the problem. [4]

In addition, the performance of the cooperative policy in the PC network is compared with the performance of the cooperative policy in the CWN by simulations, when the arrival rate of U1 varies between 1 to 6 packets per frame and $\lambda_2 = 1$. The probabilities that the channel between U2 and AP in the PC network is not available for communication are set to 10% and 30%. The simulation results show, that for the cooperative algorithm in the PC network the long-term average service rates equal to the long-term average arrival rates for all users. As the maximum supportable arrival rate of U1 in the presence of both 10% and 30% link uncertainty is 5.5 packets/frame, our dynamic policy provides very good performance even in the presence of link uncertainty between U2 and the AP. However, we omit the figure due to the lack of space.

The long-term average delays of U1 and U2 for the cooperative policy are plotted as a function of λ_1 for both PC network and CWN in Fig. 19, when $\lambda_2 = 1$. In the PC network, the probability that the channel between U2 and AP is not available for communication is 10% or 30%. It can be seen, that by adapting to changes in network conditions, our optimal policy mitigates the effect of link uncertainty providing delays only slightly longer or equal to the delays of the CWN. This is due to the fact that even if the channel between U2 and the AP is not available for communication, it is possible to use the channel between U1 and AP instead. The use of the channel between U1 and AP instead of the one between U2 and the AP results into slightly higher power

[4] *Remark*: It can be shown by simulations that the average delay is monotone non-increasing in β. Thus, in principle, it is possible to implement an iterative algorithm to compute β that keeps the delay under certain bound. Start with an arbitrary choice of β such that $\beta > 0$ and compute the optimal transmission policy and the long-term average delays for every combination of λ_1 and λ_2 inside the stability region (see Fig. 13). If the average delay of one or both users is higher (lower) than the given bound for any combination of the arrival rates, decrease (increase) the value of β and recompute. Repeat until the desired value for the delay is reached. The monotonicity property guarantees the convergence of this iteration.

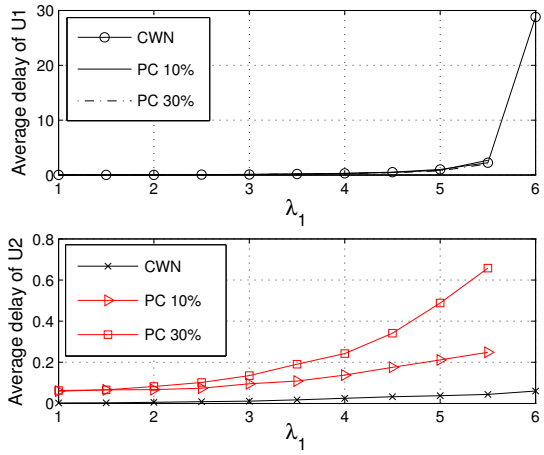

Fig. 19. Average delays for the optimal cooperative policy as a function of λ_1 for both CWN and PC network, when $\lambda_2 = 1$.

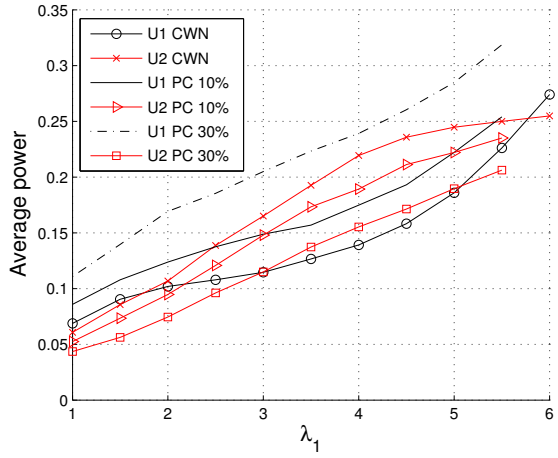

Fig. 20. Average powers for the optimal cooperative policy as a function of λ_1 for both CWN and PC network, when $\lambda_2 = 1$.

consumption in the PC network than in the CWN as can be seen in Fig. 21. Due to the uncertain link in the PC network, the delay of U2 is slightly longer in the PC network than in the CWC, as can be seen in Fig. 19.

In Fig. 20, the long-term average powers of the cooperative control policy are plotted as a function of λ_1 for PC network and CWN. It can be seen from the figure that the transmission powers of U2 for PC network are lower than the transmission power of U2

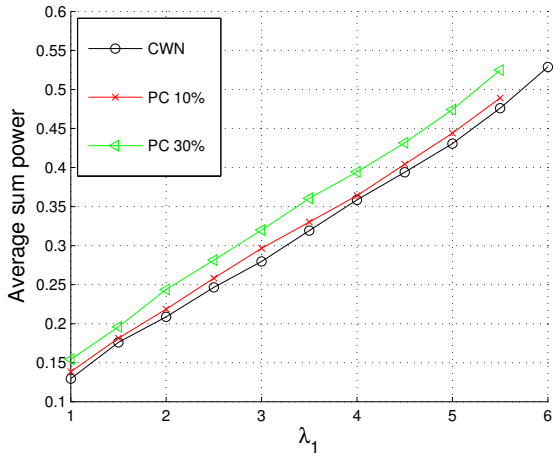

Fig. 21. Average sum powers for the optimal cooperative policy as a function of λ_1 for both CWN and PC network, when $\lambda_2 = 1$.

for CWN. This is because there are less $V_X^{ij} = 0$ and $V_X^{ij} = 3$ actions in the PC network than in the CWN, due to the uncertain link availability between U2 and the AP. However, in order to keep delay of both users as short as possible, it is possible to use the channel between U1 and the AP when the channel between U2 and the AP is not available for communication. Thus, an increase in the number of $V_X^{ij} = 1$ and $V_X^{ij} = 2$ actions increases the power usage of U1 in the PC network, as can be seen in Fig. 20.

Finally, the performance of the cooperative policy in the SSP cognitive network is compared with the performance of the cooperative policy in the CWN by simulations. The SSP cognitive network stability region is illustrated in Fig. 13, where the long-term maximum supportable arrival rates in SSP cognitive network are $p_H^{\mathrm{S}} p(I = 1)6 = 0.9 \times 0.95 \times 6 = 5.13$ packets/frame and $0.7 \times 0.95 \times 6 = 3.99$ packets/frame. The simulation results support the stability analysis in Section 5.7, since the arrival rates are equal to the service rates even on the boundary of the stability region of the SSP cognitive network. We omit the figure due to lack of space.

In Fig. 22, the average long-term delays for the cooperative policy are plotted as a function of λ_1 for both CWN and SSP cognitive network (CN). The probabilities that the channels are not available for communication in the SSP cognitive network are $p_0^{\mathrm{S}} = 0.1$. and $p_0^{\mathrm{S}} = 0.3$. It can be seen in Fig. 22, that our dynamic control policy provides bound on long-term average delays when arrival rates are inside the stability region of SSP cognitive network. Furthermore, the uncertain link availability in SSP cognitive network

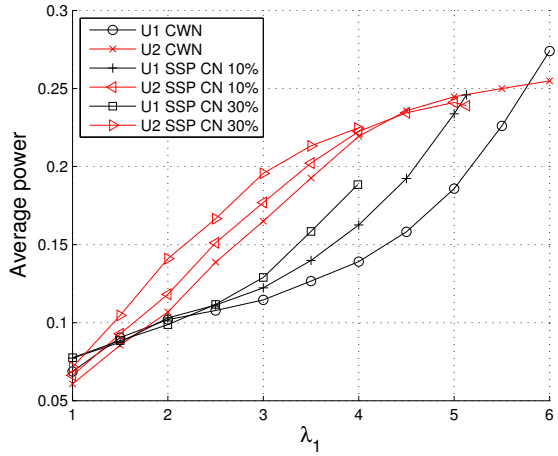

Fig. 22. Average powers for the cooperative policy as a function of λ_1 for CWN and SSP cognitive network, when $\lambda_2 = 1$.

results in only slightly higher power consumption in the SSP cognitive network than in the CWN as illustrated in Fig. 23.

4.8 Chapter summary

In this chapter, we have considered a unified model of cooperative communication network with uncertain channels and queuing for both PSP and SSP cognitive networks as well as for CWNs. For this model, we have created a unified optimization problem, where the goal was to map from the current queue states and channel gains to opportunistic cooperative control decisions that maximize the long-term average throughput of the system while maintaining queue stability. Dynamic programming methods and VIA are used to solve the unified dynamic optimization problem, and to generate an optimal control policy that maximizes the long-term average throughput.

In order to compare the potential performances of cooperative and non-cooperative communication systems to each other, we have characterized the unified capacity regions of the different networks for both PSP and SSP cognitive networks and for CWNs. The capacity region of the cooperative system was found to be 40% larger than that of the corresponding non-cooperative system. In addition, the concept of InTeNet was proposed to provide a performance upper bound for different networks.

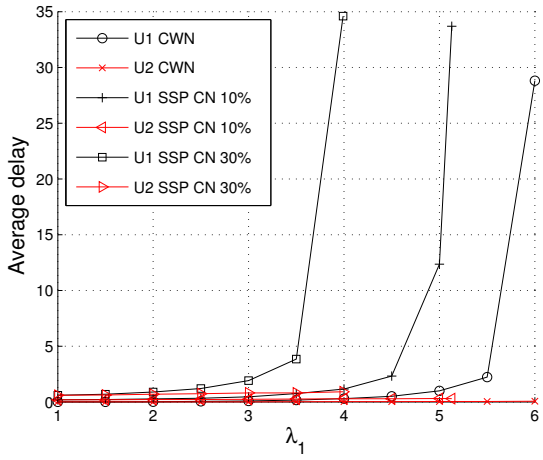

Fig. 23. Average delays for the cooperative policy as a function of λ_1 for CWN and SSP cognitive network, when $\lambda_2 = 1$.

We have presented a unified stability analysis for both PSP and SSP cognitive networks as well as for CWNs. Finally, we showed that our dynamic policy minimizes the average maximum queue length over all terminals and stabilizes the network.

Simulation results were provided to compare the performance of our dynamic cooperative policy to the performance of the corresponding non-cooperative case and to validate the theoretical analysis of this chapter. The results showed that by adapting to changes in network conditions, dynamic cooperative policy can mitigate the impact of PSP and SSP cognitive networks on each other.

The contributions of this chapter can be summarized as follows:

– A comprehensive unified cooperative network model for both PSP and SSP CNs as well as for CWN was developed.
– The model decoupled performance analysis of PSP and SSP CNs
– Two new cooperative control policies were introduced.
– We illustrated the unified stability regions for different networks.
– We showed that our optimal control policy minimizes the maximum queue length and stabilizes the network.

5 Dynamic reconfigurable wireless internet topology control and stability

In this chapter, a new paradigm in wireless network access is presented and analyzed. We consider an advanced wireless technology in which each terminal in an ad hoc or multi-hop cellular network can be turned into an AP any time it is connected to Internet. This leads to a dynamically changing topology, since the number and location of these APs can vary in time. With a slight modification of the existing technology a personal computer connected to Internet can serve as an AP, and nowadays even smart phones are designed to have such features [105]. Such technology creates a possibility that a number of potential APs can be activated to create a backhaul network and to serve a set of wireless terminals in their vicinity in an optimal way in accordance with some optimization criteria.

The goal is to control the network topology and the resource allocation in order to maximize the network performance with the minimum power consumption and to stabilize the network. The joint selection of the number of active APs and the optimal connections between the APs and the terminals in a wireless network is formulated as a dynamic optimization problem. We reformulate the dynamic problem into a MDP and use VIA to get an optimal dynamic control policy that adapts to the changes in network conditions and solves the problem. However, as the proposed optimal control policy assumes complete knowledge of the full system state information, we identify the computational complexity of the problem, and use approximate dynamic programming methods [16] and one step VIA to provide close to optimal and sub-optimal control policies that make the implementation more feasible with controllable loss in performance.

We then illustrate the network stability region and the stability regions of the optimal, close to optimal and sub-optimal policies. By comparing the stability regions of the proposed policies to each other, we can evaluate the performances of the optimal, close to optimal and sub-optimal policies.

In addition, the K-step Lyapunov drift are used to analyse the stability and the performance of the optimal control policy. We show that, if the arrival rates are inside the network stability region, our optimal control policy stabilizes the network.

Finally, the simulation results are provided in order to support the theoretical analysis of this chapter and to compare the performance of the optimal, close to optimal and sub-optimal policies to each other.

This chapter is organized as follows. In Section 5.1 the motivation behind this research and the related work is presented. Additional background on methods used in this work is presented in Section 5.2. Section 5.3 describes the system model and in Section 5.4 we formulate the optimization problem. In Section 5.5, we reformulate the dynamic problem as a MDP, and use VIA to calculate the optimal control policy. Approximate dynamic programming methods and one step VIA are used to provide suboptimal solutions in Section 5.6. The network stability region is illustrated in Section 5.7 and in Section 5.8 we compare the performance and the complexity of different algorithms to each other. In Section 5.9, we analyse the stability and the performance of the proposed dynamic control policy. The simulations are conducted to validate the theoretical analysis of the work in this chapter and presented in Section 5.10. Finally, some concluding remarks are offered in Section 5.11.

5.1 Motivation and related work

In this section, the motivation behind the research and the related work on this area are presented.

5.1.1 Motivation

Due to the fast development of portable devices equipped with advanced technology and diversified services of fixed Internet, it is desirable that mobile terminals inside the ad hoc or multihop cellular network are able to connect directly to the external networks such as Internet. In addition, as more and more advanced devices become mobile, the ability to change the point of attachment to the Internet without the need to terminate any ongoing communications will be a necessity. Such a technology has been introduced as a potential way to overcome the resource restrictions on mobile computing [52] allowing mobile users to achieve a wide variety of mobile services at low cost [59].

In this chapter, a novel approach for integrating an ad hoc or multi-hop cellular network and the Internet is presented, where a set of wireless terminals (personal computers or smart phones) in an ad hoc network can be turned into access points (APs) on demand and any time when connected to the Internet. Thus, the APs can be even

mobile. This provides substantial advantages enabling adaptation to wide range of traffic demand variations without any change in network infrastructure still offering direct access to the diversified services of Internet. Our approach reduces network design costs, since adding additional gateways is expensive in terms of hardware and labor required to install and determine the optimal configuration of the APs, which depends on the instantaneous distribution of the traffic. When the spatial traffic distribution changes, some parts of the pre-installed network infrastructure may become idle, that reduces the efficiency of the pre-investment into the network. In our solution, the network can grow and shrinks according to the changes in the spatial distribution of the traffic.

5.1.2 Related work

There are several issues that need to be taken into account when the integration of an ad hoc or multi-hop cellular network and Internet is considered. As a terminal in an ad hoc or multi-hop cellular network can freely move around, the highly dynamic topologies of such networks makes the integration procedure challenging. Since APs to Internet act as gateways between ad hoc terminals and the fixed Internet, they are likely to have heavy traffic. Therefore it is likely that these APs become bottleneck nodes if no attention is paid to enhance the availability and optimal selection of the APs.

A number of strategies to support the connectivity between ad hoc network and Internet have been proposed so far. A review of such strategies has been presented in [106]. If there are several reachable APs for a mobile terminal at some point of time, the most suitable AP can be selected according to a certain metric, e.g. strength of the received signal, number of hops between the terminal and the AP, shortest Euclidean distance [107], AP load [108], residual capacity [109], fairness [110] or combinations of some these criteria [107], [111]. A number of AP selection strategies have been proposed so far [106], [112], [113], [114], [115], [116]. Most of them are based on simple hop count approach [117], [118], where a mobile terminal chooses the AP that is closest in terms of number of physical hops. The advantages of using the shortest path selection strategy is its simplicity, low price and rapid convergence. However, if all terminals select the nearest AP, this AP can become a bottleneck. As a solution, Huang et al. [113] proposed Minimum Load Index (MLI) approach that was used to select the optimum AP based on the load of the AP. Although the MLI approach was designed to take into account the balance between the APs' loads, it does not consider loads along the path between the mobile host and the AP. In addition, creating the accurate estimate

of current load at the AP might be challenging due to the rapid fluctuations of the traffic. As a solution, the work in [119] improves the load based selection scheme by proposing an AP selection strategy based on Running Variance Metric (RVM) [115] and Relative Network Load (RNL). In [115], an AP with the least congested path to the terminal is chosen as an optimal AP based on the variance of the time that elapses between the receptions of two successive advertisement messages. A different approach for the AP selection problem was proposed in [116], where the Internet AP selection problem was formulated as a mixed integer linear program and the results were compared to the performance of the minimum hop count and load based schemes. It was shown that by formulating the AP selection problem as a linear program, significant performance improvement can be achieved compared to the conventional methods based on load, hop count and proactive routing protocols. The downside of this approach is its impracticality in real world mobile ad hoc networks. Each of the selection strategies proposed so far has their own weaknesses and benefits, thus motivating further research on the selection schemes of Internet APs using different algorithms based on some other metrics.

While some of the previous works use additional pre-installed APs [116], [120], [121], [122], others use mobile APs or the combination of the two. However, so far we have not seen any approach where the number and the location of APs can vary in time. As our dynamic optimization approach makes it possible to choose the optimal topology adaptively from the set of potential APs that the terminal can reach through the wireless links, dynamic programming is expected to lead to a stable network and better performance than the conventional AP selection methods. Since the conventional algorithms fail to adapt to the changing network conditions, they are not able to fully harvest the available capacity in the network.

5.2 Background

In this section, additional background on methods used in this chapter are presented.

5.2.1 Preclustering

To meet the increasing demand for high data rate services and to increase the energy efficiency of wireless handsets, wireless mobile operators are moving toward smaller cell/femto cell structures. However, these structures are pre-installed, expensive and not adaptive to the changes in the traffic distribution.

So as to benefit the small cell structure in practice, there is a need to mitigate the interference between collocated femtocells [123], [124], [125]. The frequency reuse factor used to mitigate the interference in macro cellular network can be useful to mitigate interference also in small cell networks. However, the reuse solutions proposed for macro cellular networks might be impractical in small cell networks. In this case, a careful attention should be paid to minimizing the number of used frequencies in the network while at the same time providing sufficient transmission quality for all terminals in the network. It was shown in [108] that the optimal solution for a joint AP selection and channel assignment problem in cellular networks in terms of minimizing the number of orthogonal channels for given load and reuse factor provides gain in blocking probability compared to other solutions. In [126], clustering is shown to significantly decrease the power consumption in wireless local area network (WLAN).

In order to simplify the optimization of large networks, we introduce the preclustering technique to segmentize the network into smaller subnetworks. Different from the fractional frequency reuse solutions proposed for macrocells, in our case, the location and the size of the clusters can also vary in time. By adapting to the changes in network conditions, the preclustering technique can jointly maximize the network performance and minimize the number of used frequencies in the network. This makes the use of frequency reuse factor practical also in small cell networks.

5.3 System model and assumptions

The dynamic network architecture (DNA) considered in this chapter is illustrated in Fig. 24. The network consists of L potential APs and N terminals. Let $\mathscr{L}$ denote the set of potential APs in the network and $\mathscr{N}$ represents the set of terminals in the network. Let U_n ($n = \{1, 2, ..., N\}$) denote the nth terminal and AP_l ($l = \{1, 2, ..., L\}$) denotes the lth AP. Time is divided into slots t, and $M(t)$ out of L potential APs are chosen to be active in each time slot. Thus, we have $M(t) \in \{0, 1, ..., L\}$. An AP is said to be active, if a terminal can use the given channel to connect to Internet via the AP. Let $\mathscr{M}(t)$ ($\mathscr{M}(t) \subset \mathscr{L}$) represent the set of active APs in slot t, $m \in \mathscr{M}(t)$ and AP_m represents mth active AP. If $M(t) = 0$, $\mathscr{M}(t) = \{\}$.

At the beginning of each slot t, fixed size packets (each of length b bits) arrive to each terminal U_n. The arrival processes $a_n(t)$ are stationary and ergodic with average

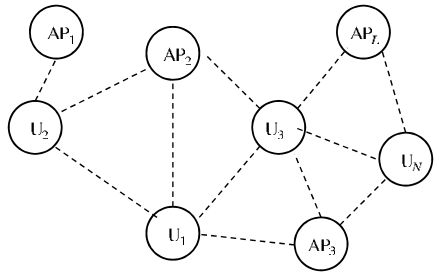

Fig. 24. System model, Dynamic Network Architecture (DNA)

rates λ_n packets/slot. Thus,

$$\lim_{t\to\infty}\frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\{a_n(\tau)\}=\lambda_n,\quad\forall\quad n\in\mathscr{N} \tag{163}$$

with probability 1 [10]. We assume that the arrivals $a_n(t)$ are bounded in their second moments every time slot, so that $\mathbb{E}\{[a_n(t)]^2\}\leq a_{\max}^2$. Let $\vec{\lambda}=[\lambda_1,...,\lambda_N]$ denote the vector of average arrival rates λ_n.

Let $\mathscr{T}$ represent the set of all possible network topologies, i.e., the set of all different connections between the terminals and the active APs. In addition, let $T(t)\in\mathscr{T}$ denote a specific network topology of (U_n,AP_m)-connections in slot t. Let $\vartheta_{nm}(t)$ denote a binary variable in time slot t given as

$$\vartheta_{nm}(t)=\begin{cases}1; & \text{If terminal } n \text{ is connected to AP}_m.\\ 0; & \text{Otherwise.}\end{cases}$$

We assume that each terminal U_n cannot be connected to more than one active AP at a time. Thus, $\sum_{m\in\mathscr{M}(t)}\vartheta_{nm}(t)\leq 1$ for each terminal U_n.

Each time slot t can be further divided into $\triangle(t)$ subslots $\hat{t}$ for scheduling on a TDMA principle. Subslot $\hat{t}=\{1,...,\triangle(t)\}$ and $\triangle(t)\in\{1,...,N\}$. One should note that the terminals connected to different APs transmit at different subslots $\triangle(t)$ according to the TDMA. However, the terminals connected to the same AP transmit simultaneously in one time slot if $N=2$ or at least in 2 subslots if $N>2$, as illustrated in Fig. 25.

Let $\mu_{nm}(t)$ denote the service rate between terminal U_n and an active access point AP$_m$ in slot t and $q_n(t)$ represents the backlog of terminal U_n in slot t. The queuing dynamics are given as

$$q_n(t+1)=q_n(t)+a_n(t)-\sum_{m\in\mathscr{M}(t)}r_{nm}(t), \tag{164}$$

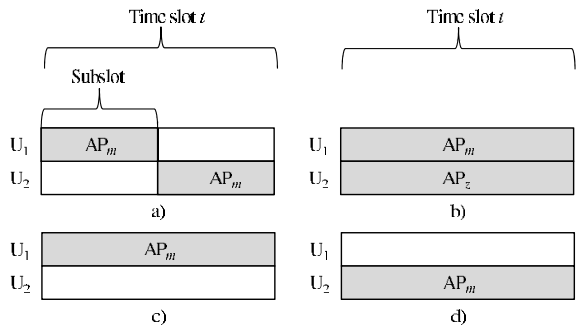

Fig. 25. Feasible time slot allocations and (U_n, AP_m)-connections for a system with 2 terminals and 2 APs in time slot t, when $M(t) \geq 1$.

where $r_{nm}(t) = \vartheta_{nm}(t)\mu_{nm}(t)$. Let $\vec{r}(t) = [\sum_{m\in\mathscr{M}(t)} r_{1m}(t), \sum_{m\in\mathscr{M}(t)} r_{2m}(t), \cdots, \sum_{m\in\mathscr{M}(t)} r_{Nm}(t)]$ and $\vec{Q}(t) = [q_1(t), q_2(t), \cdots, q_N(t)]$ represent the vectors of service rates and queue lengths at the terminals in slot t, respectively. In addition, let $y_n(t) = q_n(t) + a_n(t)$ and $\vec{Y}(t) = [y_1(t), ..., y_{|\mathscr{I}|}(t)]$ denote the vector of $y_n(t)$s.

A block fading model for the channel is assumed so that the channel gains remain fixed during a slot and change from slot to slot according to a Markov chain [5]. We use $\vec{H}_n(t) = [|h_{n1}(t)|^2, |h_{n2}(t)|^2, \cdots, |h_{nL}(t)|^2] \in \mathscr{H}_n$ to denote the vector of channel gains between terminal U_n and access point AP_l, and $\mathbf{H}(t)$ represents $N \times L$ channel gain matrix in slot t. The channel process $\mathbf{H}(t)$ is stationary and ergodic and takes values on a finite state space $\mathscr{H}$. Let π_H represent the steady state probabilities for the channel states $\mathbf{H}$.

Let $P_n(t)$ represent the power consumption of terminal U_n in time slot t, and $P^{\max}$ is the maximum power available at each terminal in time slot t. We use $\vec{P}(t) = [P_1(t), \cdots, P_N(t)]$ to denote the vector of power consumption levels at terminals.

Given $P_n(t)$ and $H_n(t)$, the service rate $\mu_{nm}(t)$, i.e., the number of packets transmitted from terminal U_n to an active access point AP_m should satisfy the following constraint

$$\mu_{nm}(t) \leq C_{nm}(t) \tag{165}$$

[5]The finite state block fading Markov chain has been widely used to model the channel in the literature, e.g. [12], [15], [20]. The model has been used to mathematically characterize Rayleigh fading channel in [86] and [87]. Using block fading model for the channel, we can dynamically generate artificial channel states that are analytically tractable and can provide closed-form results. The assumption that the channels hold their states during a slot is an approximation, which is valid for systems, whose slots are short in comparison to the channel variation. In practice, channels may vary continuously.

where

$$C_{nm}(t) \leq \frac{1}{\triangle(t)b}\log(1+\frac{|h_{nm}(t)|^2P_n(t)}{N_0+\sum_{i\in\mathscr{I},i\neq n,z\neq m}|h_{iz}(t)|^2P_i(t)}). \quad (166)$$

It is also assumed that we cannot transmit more packets than there are in the queue. In (166), $m \in \mathscr{M}(t)$, N_0 denote additive white Gaussian noise with zero mean and variance σ^2, $\mathscr{I}$ is the subset of terminals simultaneously transmitting to their own access points AP_z ($z \neq m, z \in \mathscr{M}(t)$) and $\sum_{i\in\mathscr{I},i\neq n,z\neq m}|h_{iz}|^2(t)P_i(t)$ represents overall interference generated by other terminals i simultaneously talking to its own access point AP_z. For simplicity, the service rates are restricted to integer multiples of packet lengths.

Only terminals connected to different APs can transmit simultaneously. The terminals connected to same AP transmit at different subslots $\hat{t}$ on the TDMA principle. As an example, all the feasible time slot allocations and (U_n, AP_m)-connections for a system with 2 terminals and 2 APs are illustrated in Fig. 25, where $M(t) \geq 1$, $N = L = 2$, $m \in \mathscr{M}(t)$, $z \in \mathscr{M}(t)$, $m \neq z$ and $m = z = \{1,2\}$. The grey shaded areas illustrate the parts of the time slot that are used for transmission from U_1 and/or U_2 to their APs. In Fig. 25 a), U_1 and U_2 are connected to the same AP (AP_m) and transmit in different subslots $\hat{t}$. In Fig. 25 b), both terminals transmit to different APs (U_1 transmits to AP_m and U_2 transmits to AP_z). The terminals can then use the entire time slot t for transmission. In Figs. 25 c) and 25 d), U_1 and U_2 can use the entire time slot as only one terminal is active in that slot.

5.4 Problem formulation

The goal of this work is to allocate to all N users $M(t)$ out of L ($M(t) \leq L$) APs in an optimal way in order to maximize a joint utility of the long-term average throughput of the terminals and to minimize the total power usage in the overall system while keeping the queues stable.

Let $X(t) = \{L, N, \vec{Y}(t), \mathbf{H}(t)\}$ represent the state of the system in slot t with countable state space $\mathscr{X}$. In addition, we use $W_X(t) = \{\mathscr{M}(t), r(t), \Delta(t), T(t)\}$ to denote the control input, i.e., the action, in state $X(t)$. The control input $W_X(t)$ takes values in a general state space $\mathscr{W}_X$, which represents all feasible control options in state $X(t)$. At the beginning of each slot, the network controller decides upon the value of $W_X(t)$ depending on the current $X(t)$. Starting from state X, we use $\pi = \{W_X(0), W_X(1), ...\}$ to denote a policy, i.e., sequence of actions, that in time slot t generates an action $W_X(t) \in \mathscr{W}_X$ depending upon the entire history of previously chosen state-action pairs

for $\tau = 0, 1, 2, ..., t-1$. Let Π denote the state space of all policies $\pi \in \Pi$. One should note that Π is the state space of all possible combinations of the sequences of actions $\{W_X(1), W_X(2), ...\}$, where each action $W_X(t) = \{\mathscr{M}(t), \vec{r}(t), \Delta(t), T(t)\} \in \mathscr{W}_X$ and $\mathscr{M}(t) \in \mathscr{L}$, $\mu_{nm}(t) \in \{0, 1, ..., \min\{y_n(t), C_{nm}(t)\}\}$, $\triangle(t) \in \{1, ..., N\}$ and $T(t) \in \mathscr{T}$.

The goal of this work is to map from the current $X(t)$ to an optimal sequence of $W_X(t) = \{\mathscr{M}(t), \vec{r}(t), \Delta(t), T(t)\}$, i.e., policy, that solves the following optimization problem:

$$\underset{\pi \in \Pi}{\text{maximize}} \quad \lim_{t \to \infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}_X^{\pi} \Big\{ \sum_{n \in \mathscr{N}} y_n(\tau) \sum_{m \in \mathscr{M}(\tau)} r_{nm}(\tau) - \rho M(\tau) \Big\}$$

$$\text{subject to} \quad \lim_{t \to \infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}_X^{\pi} \{P_n(\tau)\} \leq P^{\max} \quad \forall \quad n \in \mathscr{N}. \tag{167}$$

In (167), ρ represents weight that describes the relative importance of the cost of using $M(\tau)$ out of L active APs over the sum throughput. The objective in (167) encourages to allocate the largest link capacity to the terminals with the longest queues while minimizing the number of active APs in the network.

5.5 Optimal control algorithm

The control problem given in (167) is a constrained dynamic optimization problem. One way to solve it is to convert it into an unconstrained problem [41], [42], [92]. The unconstrained problem is a standard Markov Decision Process (MDP) and we define the optimal policy for this MDP using the Value iteration algorithm (VIA) [15], [95], [94]. When calculating the optimal policy, it is assumed that centralized control is possible so that the network controller has access to the full knowledge of $X(t)$.

5.5.1 Formulation as a Markov Decision Process

The set of feasible actions W_X in each state $X = \{L, N, \vec{Y}, \mathbf{H}\}$ is the set of all actions $\{\mathscr{M}, \vec{r}, \Delta, T\}$ that satisfy the power and the queue constraints as we cannot transmit more packets than there are in the queue. After taking an action W_X, the following state is given as Z. . We now let $p(Z|X, W_X)$ denote the transition probability from state X to state Z with action W_X.

For a policy $\pi \in \Pi$, we define the reward D and the power cost K by

$$D = \lim_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}_X^{\pi} \Big\{ \sum_{n\in\mathcal{N}} y_n(\tau) \sum_{m\in\mathcal{M}(\tau)} r_{nm}(\tau) - \rho M(\tau) \Big\} \tag{168}$$

$$E = \lim_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}_X^{\pi} \{ P(\tau) \} \tag{169}$$

Given the constraint in (167), let Π_E denote the set of all admissible control policies $\pi \in \Pi$ which satisfy the constraint $E \leq P^{\max}$ for each terminal. Then, (167) can be restated as a constrained optimization problem (CP) [15]:

$$\text{maximize} \;\; D \;\; \text{subject to} \;\; \pi \in \Pi_E. \tag{170}$$

The problem given in (170) can be converted into a family of unconstrained optimization problems (UP) through a Lagrangian relaxation [93]. For every $\beta_n \geq 0$, we define a corresponding Lagrangian function for any policy $\pi \in \Pi$ as,

$$J_{\beta}^{\pi}(X) = \lim_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}_X^{\pi} \Big\{ \sum_{n\in\mathcal{N}} y_n(\tau) \sum_{m\in\mathcal{M}(\tau)} r_{nm}(\tau) - \beta_n P_n(\tau) - \rho M(\tau) \Big\} \tag{171}$$

The Lagrangian multiplier β_n indicates the relative importance of power consumption over the average service rate, i.e., larger value of β_n corresponds to placing more importance on saving the transmission power at terminal U_n. We now define the unconstrained optimization problem as

$$\text{maximize} \;\; J_{\beta}^{\pi}(X) \;\; \text{subject to} \;\; \pi \in \Pi. \tag{172}$$

An optimal policy for unconstrained problem is also optimal for the original constrained control problem when $\beta = [\beta_1, ..., \beta_N]$ is appropriately chosen [15], [93].

The problem given in (172) is a standard MDP with maximum average reward criterion. For each initial state X, we define the corresponding discounted MDP with value function

$$J_{\alpha}(X) = \max_{\pi\in\Pi} \sum_{t=0}^{\infty} \mathbb{E}_X^{\pi} \left\{ \alpha^t \sum_{n\in\mathcal{N}} y_n(t) \sum_{m\in\mathcal{M}(t)} r_{nm}(t) - \beta_n P_n(t) - \rho M(t) \right\} \tag{173}$$

where the discount factor $\alpha \in (0,1)$, and a reward from taking an action $W_X(t)$ in state $X(t)$ is defined as

$$R[W_X(t), X(t)] = \sum_{n\in\mathcal{N}} \sum_{m\in\mathcal{M}(t)} y_n(t) r_{nm}(t) - \beta_n P_n(t) - \rho M(t). \tag{174}$$

$J_\alpha(X)$ is defined as the optimal total expected discounted utility for discount factor α [94].

For notational simplicity, we suppress the subscript α. The solution to (173), i.e., the optimal value functions $J^\star(X)$ for each initial state X and the corresponding discount optimal policy $\pi^\star \in \Pi$, can be solved with the following value iteration algorithm (VIA) [95]:

$$J^{l+1}(X) = \max_{W_X \in \mathscr{W}_X} \{R(W_X, X) + \alpha \sum_{Z \in \mathscr{Z}} p(Z|X, W_X) J^l(Z)\}, \tag{175}$$

In (175) $\mathscr{Z}$ is the set of feasible states that follow state X by taking an action W_X, and l denotes the iteration index. The optimal control policy, $\pi^\star$, can then be calculated by defining the optimal action $W_X \in \mathscr{W}_X$ in each state X as

$$\arg \max_{W_X \in \mathscr{W}_X} \left\{ R(W_X, X) + \alpha \sum_{Z \in \mathscr{Z}} p(Z|X, W_X) J^\star(Z) \right\}. \tag{176}$$

Dynamic control algorithms has been used in several publications such as [9], [12], [15], [95], and the optimal control policy can in principle be calculated using the exact dynamic programming methods as described above. However, as our optimal control policy assumes complete knowledge of the available active APs, channel state information (CSI) and queue state information (QSI), the cost of implementing exact dynamic programming increases, when the number of states grows. Thus, the optimization process will be effective only for relatively small L, M and N. We simplify the signal processing by assuming that a large ad hoc or multi-hop cellular network can be segmented into small clusters and use frequency reuse factor to eliminate the intercluster interference, as described in Subsubsection 5.2.1. Thus, the above performance optimization can be done separately for each subnetwork. In this work, we assume a priori clustering and leave out any further details regarding this issue.

5.6 Approximate solutions

Dynamic optimization problems can in principle be solved using exact dynamic programming [95]. However, exact dynamic programming approach has been found to be intractable for many problems in practice, due to the so-called "curse of dimensionality". Separable dynamic optimization problems [127] can be efficiently solved by obtaining additive separable approximations of the optimal value function. Due to the computational complexity of the large network control problems using the exact

dynamic programming methods, approximate dynamic programming [127] can be used to provide less complex close to optimal and sub-optimal policies. In this section, we simplify the close to optimal policy presented in [15] and use approximate dynamic programming tools to derive two new suboptimal approximate policies to solve (167).

5.6.1 Close to Optimal Policy

In [15], the authors approximated the optimal value functions by decomposing the right hand side of (175) into additive terms each one being a function of only one user's variable and then applying one step VIA to obtain a close to optimal solution for an optimal dynamic optimization problem.

Let $x_n(t) = \{L, N, y_n(t), \mathbf{H}(t)\}$ represent the state at terminal U_n in slot t with countable state space $\mathscr{X}_n$. We use $w_n^x = \{\mathscr{M}_n(t), \Delta_n(t), \sum_{m\in\mathscr{M}_n(t)} r_{nm}(t), T_n(t)\}$ to represent an action at terminal U_n in state $x_n(t)$. The action $w_n^x(t)$ takes values on a state space $\mathscr{W}_n^x$, where $\mathscr{M}_n(t) \in \mathscr{L}$, $T_n(t) \in \mathscr{T}$, $\mu_{nm}(t) \in \{0, 1, ..., \min\{y_n(t), C_{nm}(t)\}\}$ and $\Delta_n(t) \in \{1, ..., N\}$. We use π_n to denote the sequence of actions at terminal U_n, and let Π_n be the state space of all such policies.

In order to decompose (175), we assume that the simultaneous transmissions see the maximum possible interference, i.e. all simultaneously transmitting terminals in time slot t transmit to different APs with the highest power possible $P^{\max}$ [15]. The problem in (173) can now be decomposed terminal-wise as

$$\underset{\pi_n\in\Pi_n}{\text{maximize}} \quad \mathbb{E}_{x_n}^{\pi_n}\left\{\sum_{t=0}^{\infty}\alpha^t \sum_{m\in\mathscr{M}_n(t)} y_n(t) r_{nm}(t) - \beta_n P_n(t) - \rho\frac{M_n(t)}{N}\right\}, \tag{177}$$

where $r_{nm}(t) = \vartheta_{nm}(t)\mu_{nm}(t)$ and $\mu_{nm}(t)$ are obtained by replacing $\Delta(t)$ by $\Delta_n(t)$ and $P_i(t)$ by $P^{\max}$ in (166). Note that $\sum_{i\in\mathscr{I}, i\neq n, z\neq k} |h_{iz}(t)|^2 P^{\max}$ is the total interference caused by the simultaneous transmissions of other terminals each transmitting at maximum allowable power $P^{\max}$ to different APs.

The additive decomposed value function approximations $J_n^\star(x_n)$ can now be calculated from

$$J_n^{l+1}(x_n) = \max_{w_n^x\in\mathscr{W}_n^x}\{R_n(w_n^x, x_n) + \alpha \sum_{z_n\in\mathscr{Z}_n} p(z_n|x_n, w_n^x) J_n^l(z_n)\}, \tag{178}$$

where

$$R_n[w_n^x(t), x_n(t)] = [\sum_{m\in\mathscr{M}(t)} y_n(t) r_{nm}(t) - \beta_n P_n(t)] - \rho\frac{M_n(t)}{N}, \tag{179}$$

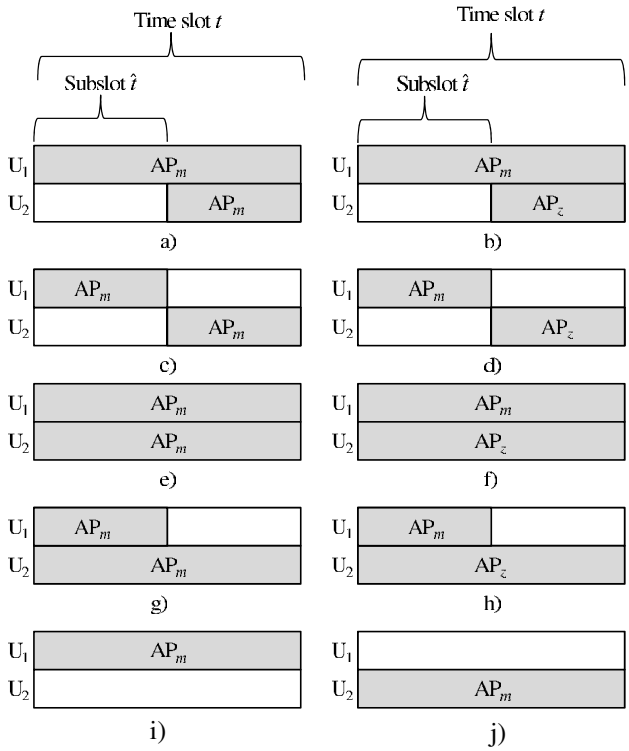

Fig. 26. All the possible time slot allocations and (U_n, AP_m)-connections of suboptimal policy 1 for a system with 2 terminals and 2 APs in time slot t, when $M(t) \geq 1$, $m = z = \{1,2\}$ and $m \neq z$.

and $\mathscr{Z}_n$ is the set of feasible states that can follow state x_n by taking an action w_n^x. The transition probability from state x_n to state z_n with action w_n^x is $p(z_n|x_n, w_n^x)$.

We replace J^l in (175) with the additive decomposed approximations given by (178) for each terminal U_n, and carry out one step VIA to calculate the close to optimal approximated value function $\widehat{J}^\star$. For each state $X \in \mathscr{X}$, the close to optimal actions W_X for the close to optimal policy are then given by

$$\arg \max_{W_X \in \mathscr{W}_X} \{D(W_X, X) + \alpha \sum_{Z \in \mathscr{Z}} p(Z|X, W_X) \widehat{J}^\star(Z)\}. \tag{180}$$

It can be seen that the above close to optimal policy first presented in [15] simplifies the dynamic optimization problem, by calculating the value functions distributedly for each terminal U_n. However, just like in the optimal case, the close to optimal actions

for each state $X \in \mathscr{X}$ are calculated centralized using the full state space information from (180). Thus, the close to optimal policy is still computationally very complex, when $\mathscr{X}$ is large. It is clear, that in order to significantly decrease the computational complexity of the dynamic optimization problem in (167), it is necessary to decrease the total number of states for which we need to calculate the action. Next, we further simplify the close to optimal solution by deriving two new suboptimal approximate policies to solve (167).

5.6.2 Suboptimal Policy 1

In this subsection, we propose a new sub-optimal control policy 1, where the approximated value functions $J_n^\star(z_n)$ are first solved distributedly from (178). We then use VIA to calculate the sub-optimal policy 1 terminal-wise by defining the best action $w_n^x = \{\mathscr{M}_n, \sum_{m\in\mathscr{M}_n} r_{nm}, \Delta_n, P_n, T_n\}$ for each state $x_n = \{L, N, y_n, \mathbf{H}\} \in \mathscr{X}_n$ from

$$\arg \max_{w_n^x \in \mathscr{W}_n^x} \{D_n(w_n^x, x_n) + \alpha \sum_{z_n \in \mathscr{Z}_n} p(z_n|x_n, w_n^x) J_n(z_n)\}. \tag{181}$$

We can now check the actions w_n^x for each terminal U_n and get the specific network topology T for each state $X = \{L, N, \vec{Y}, \mathbf{H}\} \in \mathscr{X}$, where $\vec{Y} = [y_1, \cdots, y_N]$. Let $W_n^X = \{w_1^x, \cdots, w_N^x\}$ denote the sequence of actions given by (181) in state $X \in \mathscr{X}$. The maximum service rate between terminal U_n and an active access point AP_m is now obtained by replacing Δ by Δ_n in (166). Thus, suboptimal policy 1 requires full CSI but no QSI of other terminals in the network.

Since the suboptimal actions w_n^x are calculated separately for each terminal U_n, terminals do not have any information of the connections of other terminals in the network, e.g. terminal U_n cannot not know if other terminals are connected to the same/different APs in the network. Then, $\triangle_n$ and T_n are only assumptions made by terminal U_n in state x_n that might not be true, as other terminals might assume different $\triangle_n$ and T_n. Thus, the set of all possible time slot allocations and (U_n, AP_m)-connections is different from that in Fig. 25. When terminals do not have any information of the actions of other terminals in the network it is possible that one or more terminals transmit simultaneously to the same AP and the transmissions fail. As an example, let us consider all combinations of $(\triangle_1, \triangle_2)$ and (T_1, T_2) of suboptimal policy 1 for a system with 2 terminals and 2 APs in time slot t illustrated in Fig. 26, when $M(t) \geq 1$, $m = z = \{1, 2\}$ and $m \neq z$. In Figs. 26 a) and 26 b) $\triangle_1 = 1$ and $\triangle_2 = 2$, i.e., U_1 assumes that U_2 transmits to different AP in time slot t and U_2 assumes that U_1 transmits to

same AP. Thus, terminal 1 uses the entire time slot t and terminal 2 uses only half of the time slot t (shown as shaded grey areas in Figs. 26 a) and 26 b)). In Fig. 26 b) both terminals are successfully transmitting to different APs. However, in Fig. 26 a), U_1 and U_2 transmit simultaneously to same AP_m for half time slot. Thus, the whole transmission of U_2 and half of the transmission of U_1 fail. In Figs. 26 c) and 26 d) $\triangle_1 = 2$ and $\triangle_2 = 2$ and both U_1 and U_2 assume that other terminal is connected to same AP in time slot t. Both terminals then use only half of the time slot t, but the transmissions are successful and do not interfere. In Figs. 26 e) and 26 f) $\triangle_1 = 1$ and $\triangle_2 = 1$, and both terminals assume that the other terminal is connected to a different AP in time slot t. Then, both terminals use the entire time slot for transmission, but in Fig. 26 e) both terminals transmit simultaneously to the same AP_m and the transmission fail. In Figs. 26 g) and 26 h) $\triangle_1 = 2$ and $\triangle_2 = 1$, and terminal 1 assumes that terminal 2 transmits to the same AP in time slot t and terminal 2 assumes that terminal 1 transmits to different AP in time slot t. However, due to the simultaneous transmission in Fig. 26 g), the whole transmission of U_1 and half of the transmission of U_2 fail. In Figs. 26 i), 26 j), 26 k) and 26 l), U_1/U_2 assumes that the other terminal transmits to different/same AP, when it happens that another terminal is not connected to any AP.

The proposed suboptimal policy 1 significantly reduces the computational complexity of the problem. However, the downside of the policy is the fact that since the terminals do not have any knowledge of the actions of other terminals, collisions might happen.

5.6.3 Suboptimal policy 2

In this section, we eliminate the possibility for collision from suboptimal policy 1, and propose another simplified scheme called suboptimal policy 2. Like in Section 5.5, dynamic optimization methods are used to calculate the value functions and the best actions, but decrease the state space by calculating the best actions only for a terminal with the longest queue and then use these results to approximate the actions of other terminals in the network.

By noting that the optimal control policy is designed to give priority to a terminal with the longest queue, we define $y_{n_{\max}}(t) = \max_{n \in \mathcal{N}}\{y_n(t)\}$ as the maximum backlog in time slot t over all users. In addition, let $n_{\max}$ denote the index of the terminal with the longest queue and $U_{n_{\max}}(t)$ represents the terminal with the longest queue in slot t.

Let $x_{n_{\max}}(t) = \{L, N, y_{n_{\max}}(t), \mathbf{H}(t)\}$ represent the system state in slot t with countable state space $\mathscr{X}_{n_{\max}}$. One should note that $U_{n_{\max}}(t)$ has full knowledge of N, L

and CSI but $U_{n_{\max}}(t)$ has only its own queue length information $y_{n_{\max}}(t)$. The action at terminal $U_{n_{\max}}(t)$ in state $x_{n_{\max}}(t)$ is given as $w^x_{n_{\max}}(t) = \{\mathscr{M}_{n_{\max}}(t),\ \Delta_{n_{\max}}(t),\ \sum_{m\in\mathscr{M}_{n_{\max}}(t)} r_{n_{\max}m}(t),\ T_{n_{\max}}(t)\}$. The action $w^x_{n_{\max}}(t)$ takes values on a state space $\mathscr{W}^x_{n_{\max}}$, where $\mathscr{M}_{n_{\max}}(t) \in \mathscr{L}$, $T_{n_{\max}}(t) \in \mathscr{T}$, $\sum_{m\in\mathscr{M}(t)} r_{n_{\max}m}(t) \in \{0,1,...,y_{n_{\max}}(t)\}$ and $\Delta_{n_{\max}}(t) \in \{1,...,N\}$. We use $\pi_{n_{\max}} = \{w^x_{n_{\max}}(0),\ w^x_{n_{\max}}(1),...\}$ to denote the sequence of actions, and let $\Pi_{n_{\max}}$ be the state space of all such policies.

Given $\sum_{m\in\mathscr{M}_{n_{\max}}(t)} r_{n_{\max}m}(t)$, the service rates of other terminals $U_n, n \in \mathscr{N}, n \neq n_{\max}$, are approximated as

$$\sum_{m\in\mathscr{M}_{n_{\max}}(t)} r_{nm}(t) = \min\left\{C_{nm}(t), y_{nm}(t), \frac{y_n(t)}{y_{n_{\max}}(t)} \sum_{m\in\mathscr{M}_{n_{\max}}(t)} r_{n_{\max}m}(t)\right\}, \tag{182}$$

where $v_{nm}(t) = 0$, if the channel gain between terminal U_n $(n \neq n_{\max})$ and the access point AP_m is smaller than a predetermined threshold v. Now the maximum number of packets that can be transmitted from other terminals U_n $(n \in N, n \neq n_{\max})$ are smaller than or equal to $\sum_{m\in\mathscr{M}_{n_{\max}}(t)} r_{n_{\max}m}(t)$, i.e., $\sum_{m\in\mathscr{M}_{n_{\max}}(t)} r_{nm}(t) \leq \sum_{m\in\mathscr{M}_{n_{\max}}(t)} r_{n_{\max}m}(t)$, where $n \in N$ and $n \neq n_{\max}$.

We can separate (175) by noting that all terminals simultaneously transmitting to different APs has the maximum power of $P_n[h_{nm}(t), r_{n_{\max}m}(t)]$, and the problem in (173) can now be written terminal-wise as

$$\underset{\pi_{n_{\max}}\in\Pi_{n_{\max}}}{\text{maximize}} \quad \mathbb{E}^{\pi_{n_{\max}}}_{x_{n_{\max}}}\left\{\sum_{t=0}^{\infty}\alpha^t \sum_{m\in\mathscr{M}_{n_{\max}}(t)} y_{n_{\max}}(t) r_{n_{\max}m}(t) - \beta_n P_{n_{\max}m}(t) - w\frac{M_{n_{\max}}(t)}{N}\right\}, \tag{183}$$

where $r_{n_{\max}m}(t) = \vartheta_{n_{\max}m}(t)\mu_{n_{\max}m}(t)$ and

$$\mu_{n_{\max}m}(t) \leq \frac{1}{b\Delta_{n_{\max}}(t)} \log\left(1 + \frac{|h_{n_{\max}m}(t)|^2 P_{n_{\max}}(t)}{N_0 + \sum_{i\in\mathscr{I}, i\neq n_{\max}, z\neq m} |h_{iz}(t)|^2 P_i[h_{iz}(t), r_{n_{\max}m}(t)]}\right). \tag{184}$$

Note that $\sum_{i\in\mathscr{I}, i\neq n_{\max}, z\neq m} |h_{iz}(t)|^2 P_i[h_{iz}(t), r_{n_{\max}m}(t)]$ is the total interference caused by the simultaneous transmissions of other terminals each using power $P_i[h_{nm}(t),\ r_{n_{\max}m}(t)]$

when transmitting its data to different APs. Thus, we always have

$$\log\left(1+\frac{|h_{n_{\max}m}(t)|^2P_{n_{\max}}(t)}{N_0+\sum_{i\in\mathscr{I},i\neq n_{\max},z\neq m}|h_{iz}(t)|^2P_i[h_{iz}(t),r_{n_{\max}m}(t)]}\right)$$
$$\leq \log(1+\frac{|h_{n_{\max}m}(t)|^2P_{n_{\max}}(t)}{N_0+\sum_{i\in\mathscr{I},i\neq n_{\max},z\neq m}|h_{iz}(t)|^2P_i(t)}). \quad (185)$$

The value functions for each $x_{n_{\max}}\in X_{n_{\max}}$ can now be calculated from

$$J^{l+1}_{n_{\max}}(x_{n_{\max}}) = \max_{w^x_{n_{\max}}\in\mathscr{W}^x_{n_{\max}}}\{R_{n_{\max}}(w^x_{n_{\max}},x_{n_{\max}})+$$
$$\alpha\sum_{z_{n_{\max}}\in\mathscr{Z}_{n_{\max}}}p(z_{n_{\max}}|x_{n_{\max}},w^x_{n_{\max}})J^l_{n_{\max}}(z_{n_{\max}})\}, \quad (186)$$

where

$$R_{n_{\max}}[w^x_{n_{\max}}(t),x_{n_{\max}}(t)] = [\sum_{m\in\mathscr{M}_{n_{\max}}(t)}y_{n_{\max}}(t)r_{n_{\max}m}(t)-$$
$$\beta_nP_{n_{\max}}(t)]-w\frac{M_{n_{\max}}(t)}{N}, \quad (187)$$

and $\mathscr{Z}_{n_{\max}}$ is the set of feasible states that can follow state $x_{n_{\max}}$ by taking an action $w^x_{n_{\max}}$ in state $x_{n_{\max}}$. The transition probability from state $x_{n_{\max}}$ to state $z_{n_{\max}}$ with action $w^x_{n_{\max}}$ is $p(z_{n_{\max}}|x_{n_{\max}},w^x_{n_{\max}})$.

The sequence of actions at terminal $U_{n_{\max}}$ can then be calculated by defining the best action $w^x_{n_{\max}}\in\mathscr{W}^x_{n_{\max}}$ in each state $x_{n_{\max}}$ as

$$\arg\max_{w^x_{n_{\max}}\in\mathscr{W}^x_{n_{\max}}}\Big\{D_{n_{\max}}(w^x_{n_{\max}},x_{n_{\max}})+$$
$$\alpha\sum_{z_{n_{\max}}\in\mathscr{Z}_{n_{\max}}}p(z_{n_{\max}}|x_{n_{\max}},w^x_{n_{\max}})J^\star_{n_{\max}}(z_{n_{\max}})\Big\}. \quad (188)$$

Based on $\pi_{n_{\max}}$ given by (188), the action in state $X\in\mathscr{X}$ can be now given as $W^X_{n_{\max}}=\{\mathscr{M}_{n_{\max}},\Delta_{n_{\max}},\vec{r}_{\max},T_{n_{\max}}\}$, where $\vec{r}_{\max}$ is the vector of service rates given by (182) and (188).

In order to compare the performance of the proposed policies, we first introduce the network stability region in Section 5.7. We then compare the performance and the complexity of different policies to each other in Section 5.8.

5.7 Achievable rates

The network stability region includes all input rates λ_n that the network can stably support, considering all possible resource allocation policies that we can have for the

network. In this section, we characterize the fundamental throughput limitations and establish the stability region of the proposed DNA. It is important to note that stability region is unique for each network and it should be distinguished from the stability region of a specific resource allocation policy that is a subset of the network stability region [12]. The terms network capacity region and network stability region might be used interchangeably in the text.

5.7.1 Network Stability Region

Let $\mathscr{W}_H$ denote the set of all possible resource allocation options in channel state $\mathbf{H}$, and $W_H \in \mathscr{W}_H$ represents a control action in $\mathbf{H}$. In addition, we use $\vec{G} = [g_1, g_2, \cdots, g_N]$ to denote the vector of average long-term supportable service rates at the terminals. Due to the time varying system state conditions, $\vec{G}$ must be averaged over all possible channel states. Moreover, $\vec{G}$ is not fixed and depends on transmission policy for choosing W_H. Thus, instead of describing the network with a single $\vec{G}$, the network is described as a following set of supportable service rates

$$\Gamma = \sum_{\mathbf{H}\in\mathscr{H}} \pi_H \text{Conv}\{\vec{r}(W_H, \mathbf{H}) | W_H \in \mathscr{W}_H\}, \tag{189}$$

where addition and scalar multiplication of sets is used, $\text{Conv}\{\Upsilon_H\}$ represents convex hull of the set Υ_H that is defined as the set of all convex combinations $p_1 b_1 + p_2 b_2 + ... + p_j b_j$ of elements $b_j \in \Upsilon_H$ and p_js are probabilities summing to 1. Specifically, the throughput region Γ can be viewed as a set of all long-term average service rates $\vec{G}$ that the network can be configured to support on the wireless links connecting the terminals and the APs.

The network stability region is then given as the set of all arrival rate vectors $\vec{\lambda}$ for which there exists a transmission policy that satisfies

$$\vec{\lambda} \leq \lim_{t\to\infty} \frac{1}{t} \sum_{\tau=1}^{t-1} \mathbb{E}\{\vec{r}(\tau)\} \leq \sum_{\mathbf{H}\in\mathscr{H}} \pi_H \vec{G}_H \tag{190}$$

for some $\vec{G} \in \Gamma$ where $\vec{G} = \sum_{\mathbf{H}\in\mathscr{H}} \pi_H \vec{G}_H$ for some set of average transmission rates $\vec{G}_H$ in channel state $\mathbf{H}$. The arrival rate vector $\vec{\lambda}$ is in the region Λ if there exists a long-term average rate vector $\vec{G} \in \Gamma$ such that there exists a transmission policy which supports the arrival rates $\vec{\lambda}$.

Due to the computational complexity of the network stability region in (189), Γ is calculated for a system with 2 terminals and 2 APs for the channel conditions given in Section 5.10, and illustrated in Fig. 27.

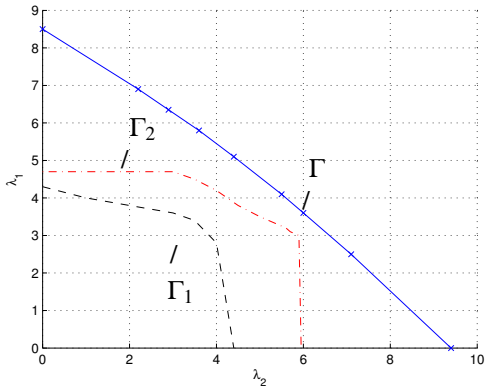

Fig. 27. Network stability region and the stability regions of the different control policies.

5.8 Performance and complexity comparison

In this section, we illustrate the performance of the optimal, close to optimal and sub-optimal policies and compare the performance of the different policies to each other.

5.8.1 Complexity

It is easy to see that the complexity of a dynamic control policy highly depends on the state space size, since the best/optimal action is calculated for each possible state in the state space.

We first define the sizes of $\mathscr{X}$, $\mathscr{X}_n$ and $\mathscr{X}_{n_{\max}}$. Let $|\mathscr{X}|$, $|\mathscr{X}_n|$ and $|\mathscr{X}_{n_{\max}}|$ denote the number states in $\mathscr{X}$, $\mathscr{X}_n$ and $\mathscr{X}_{n_{\max}}$, respectively. In addition, let $|\mathscr{H}|$ denote the number of channel states in state space $\mathscr{H}$. For arrival rates inside the stability region, $y_n(y)$ gets values between 0 and $y^{\max}$ so that

$$\limsup_{t\to\infty} y_n(t) \leq y^{\max} \tag{191}$$

for each terminal U_n. For arrival rates outside Γ, $y_n(t) \to \infty$. Then,

$$|\mathscr{X}| = N(y^{\max}+1)|\mathscr{H}| \tag{192}$$

and

$$|\mathscr{X}_n| = |\mathscr{X}_{n_{\max}}| = (y^{\max}+1)|\mathscr{H}|. \tag{193}$$

Both the value functions and the actions of the optimal control policy are calculated for $|\mathscr{X}|$ states from (175) and (176), respectively. For close to optimal policy, the approximated value functions in (178) are calculated separately for $|\mathscr{X}_n|$ states, but the actions are calculated for $|\mathscr{X}|$ states from (180). The value functions and the actions for sub-optimal policy 1 are calculated for $|\mathscr{X}_n|$ states from (178) and (181), respectively. For sub-optimal policy 2, the value functions and the actions are calculated for $|\mathscr{X}_{n_{\max}}|$ states from (186) and (188), respectively.

Sub-optimal policies are considerably less complex than close to optimal and optimal policies. The fact that the sub-optimal policies do not require full QSI decreases significantly the computational complexity of the problem. In addition, in order to further decrease the complexity of the proposed policies, parameters N and L can be reduced by using the preclustering as described in Section 5.2.1.

5.8.2 Performance

We compare the performance of the proposed policies by comparing their stability regions to each other. In Section 5.9, we prove that our optimal policy achieves every point on the network stability region. Thus, the network stability region Γ illustrated in Fig. 27 represents the stability region of the optimal control policy. In addition, we approximate the stability regions of the sub-optimal policies by simulations. Let Γ_1 and Γ_2 represent the stability regions of sub-optimal policies 1 and 2, respectively. By comparing Γ and Γ_2 in Fig. 27, it can be seen that sub-optimal policy 2 not only simplifies the computational complexity of the problem but also provides performance comparable to the performance of the optimal control policy. However, for sub-optimal policy 1, terminals do not have any knowledge of the actions of other terminals and collisions might happen. That is why Γ_1 is smaller than Γ_2. One should also note that sub-optimal policy 1 assumes that other terminals in the network transmit with maximum available power and sub-optimal policy 2 assumes that other terminals in the network transmit at rate equal to $\sum_{m\in\mathscr{M}_{n_{\max}}(t)} r_{nm}(t)$. Thus, the difference between Γ_1, Γ_2 and Γ when either λ_1 or λ_2 is small is significantly larger than the difference between Γ_1, Γ_2 and Γ for larger values of λ_1 and λ_2.

5.9 Stability analysis

In this section, we analyse the stability and the performance of the proposed optimal dynamic control policy using the K-step Lyapunov drift. Our optimal policy is shown to stabilize the network.

5.9.1 K-step Lyapunov Drift

Consider the K-step dynamics for unfinished work at terminal U_n:

$$q_n(t_0+(\hat{j}+1)K) = q_n(t_0+\hat{j}K) + \sum_{\tau=t_0+\hat{j}K}^{t_0+(\hat{j}+1)K-1} a_n(\tau) - \sum_{\tau=t_0+\hat{j}K}^{t_0+(\hat{j}+1)K-1} \sum_{m\in\mathscr{M}(\tau)} r_{nm}(\tau), \quad (194)$$

where $t_0 \in \{0,1,...,K\}$ and $\hat{j} \in \{0,1,...,\hat{J}-1\}$. We can write (194) as

$$q_n(t_0+(\hat{j}+1)K) = y_n(t_0+\hat{j}K) + \sum_{\tau=t_0+\hat{j}K+1}^{t_0+(\hat{j}+1)K} a_n(\tau) - \sum_{\tau=t_0+\hat{j}K}^{t_0+(\hat{j}+1)K-1} \sum_{m\in\mathscr{M}(\tau)} r_{nm}(\tau), \quad (195)$$

where $y_n(t_0+\hat{j}K) = q_n(t_0+\hat{j}K) + a_n(t_0+\hat{j}K)$. Adding $a_n(t_0+(\hat{j}+1)K)$ on the both sides of (195), we get

$$y_n(t_0+(\hat{j}+1)K) = y_n(t_0+\hat{j}K) + \sum_{\tau=t_0+\hat{j}K+1}^{t_0+(\hat{j}+1)K} a_n(\tau) - \sum_{\tau=t_0+\hat{j}K}^{t_0+(\hat{j}+1)K-1} \sum_{m\in\mathscr{M}(\tau)} r_{nm}(\tau), \quad (196)$$

where $y_n(t_0+(\hat{j}+1)K) = q_n(t_0+(\hat{j}+1)K) + a_n(t_0+(\hat{j}+1)K)$. Inserting $y_n = y_n(t_0+\hat{j}K)$, $\sum_{m\in\mathscr{M}} r_{nm} = \frac{1}{K}\sum_{\tau=t_0+\hat{j}K}^{t_0+(\hat{j}+1)K-1} \sum_{m\in\mathscr{M}(\tau)} r_{nm}(\tau)$, $a_n = \frac{1}{K}\sum_{\tau=t_0+\hat{j}K+1}^{t_0+(\hat{j}+1)K} a_n(\tau)$ into (196), squaring both sides of (196), defining the Lyapunov function as $L(y) = y_n^2$ and taking conditional expectation, (196) can be written as

$$\begin{aligned} &\mathbb{E}\{L[y(t_0+(\hat{j}+1)K)] - L[y(t_0+\hat{j}K)] | y_n(t_0+\hat{j}K)\} = \quad (197)\\ &\quad K^2V - 2y_n(t_0+\hat{j}K)\Big[\mathbb{E}\{\sum_{\tau=t_0+\hat{j}K}^{t_0+(\hat{j}+1)K-1} \sum_{m\in\mathscr{M}(\tau)} r_{nm}(\tau) | y_n(t_0+\hat{j}K)\} - \\ &\quad \mathbb{E}\{\sum_{\tau=t_0+\hat{j}K+1}^{t_0+(\hat{j}+1)K} a_n(\tau) | y_n(t_0+\hat{j}K)\}\Big], \end{aligned}$$

where

$$V \triangleq (\mu_{\max}^{\text{out}})^2 + (a_{\max})^2, \quad (198)$$

and $\mu_{\max}^{\text{out}}$ is the maximum transmission rate out of a given terminal U_n given as

$$\mu_{\max}^{\text{out}} \triangleq \max_{\{n\in\mathcal{N},\mathbf{H}\in\mathscr{H},W_H\in\mathscr{W}_H\}} \mu_{nm}(W_H,\mathbf{H}). \tag{199}$$

Since $\mu_{nm}(W_H,\mathbf{H})$ is bounded, $\mu_{\max}^{\text{out}}$ exists [12], [10].

The inequality in (197) was first presented in [20], and it represents the K-step Lyapunov drift for any resource allocation policy that we can have for the network.

5.9.2 Network Stabilizing Policy

In this subsection, we analyse the stability and the performance of our optimal dynamic control policy. We show that, if the arrival rates are inside the network stability region, our dynamic transmission policy stabilizes the network.

Specifically, our dynamic policy is designed to maximize

$$\lim_{t\to\infty}\frac{1}{t}\sum_{\tau=0}^{t-1}\sum_{n\in\mathcal{N}}\mathbb{E}\{y_n(\tau)\sum_{m\in\mathscr{M}(\tau)}r_{nm}(\tau)-\beta_n p_n(\tau)-\rho M(\tau)\}. \tag{200}$$

In addition, by summing

$$y_n(t_0+\hat{j}K)\mathbb{E}\{\sum_{\tau=t_0+\hat{j}K}^{t_0+(\hat{j}+1)K-1}\sum_{m\in\mathscr{M}(\tau)}r_{nm}(\tau)|y_n(t_0+\hat{j}K)\} \tag{201}$$

on the right hand side of (197) over $t_0\in\{0,1,...,K\}$, (201) can be rewritten as

$$\begin{aligned} &y_n(t_0+\hat{j}K)\mathbb{E}\{\sum_{\tau=t_0+\hat{j}K}^{t_0+(\hat{j}+1)K-1}\sum_{m\in\mathscr{M}(\tau)}r_{nm}(\tau)|y_n(t_0+\hat{j}K)\}= \\ &\sum_{t_0=0}^{K-1}y_n(t_0+\hat{j}K)\mathbb{E}\{\sum_{\tau=t_0+\hat{j}K}^{t_0+(\hat{j}+1)K-1}\sum_{m\in\mathscr{M}(\tau)}r_{nm}(\tau)|y_n(t_0+\hat{j}K)\}= \\ &\sum_{t_0=0}^{K-1}y_n(t_0+\hat{j}K)\Big[\mathbb{E}\{\sum_{m\in\mathscr{M}(\tau)}r_{nm}(t_0+\hat{j}K)|y_n(t_0+\hat{j}K)\}+ \\ &\mathbb{E}\{\sum_{\tau=t_0+\hat{j}K+1}^{t_0+(\hat{j}+1)K-1}\sum_{m\in\mathscr{M}(\tau)}r_{nm}(\tau)|y_n(t_0+\hat{j}K)\}\Big] \end{aligned} \tag{202}$$

It now is now easy to see that the dynamic policy maximizes the right hand side of (197). Thus, if the arrival rates are inside the network stability region, dynamic policy stabilizes the network and minimizes the bound for average queue length over all terminals.

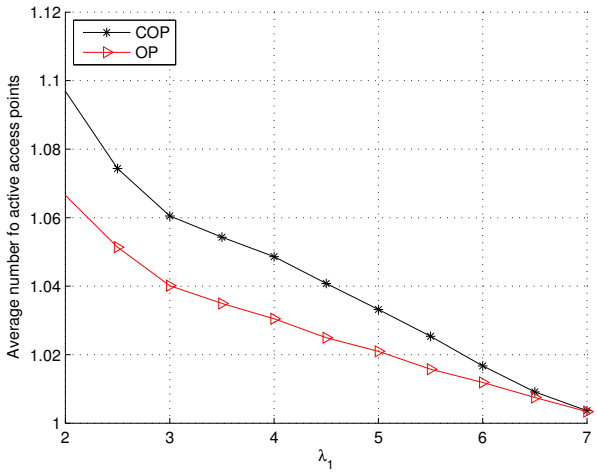

Fig. 28. Average number of active APs for optimal and close to optimal policies versus λ_1, when $\lambda_2 = 2$.

5.10 Performance evaluation

For illustration purposes, we evaluate the performance of the optimal, close to optimal and suboptimal transmission policies with simulations. Although the suboptimal schemes can be calculated and the network stability has been proven theoretically for any size of the network, due to the computational complexity of the optimal and close to optimal solutions for large networks, we introduce preclustering of the network and segmentize the network into smaller subnetworks, where $N = L = 2$. Since the interference between the subnetworks is eliminated using the given frequency reuse factor, the optimization can be performed separately for each subnetwork. The resulting power, delay and throughput curves of the optimal control policy are compared to the performance of other solutions. The results can be used to validate our stability analysis presented in Sections 5.7 and 5.9.

The channel process is generated according to a Markov chain and the stationary probabilities $p\{\mathbf{H} = (h_{11}, h_{12}; h_{21}, h_{22})\}$ of the 16 x 16 channel state transition matrix are given as $p\{\mathbf{H} = (0.1,\ 0.1;\ 0.1,\ 0.1)\} = 0.025$, $p\{\mathbf{H} = (0.1,\ 0.1;\ 0.1,\ 1)\} = 0.1$, $p\{\mathbf{H} = (0.1,\ 0.1;\ 1,\ 0.1)\} = 0.075$, $p\{\mathbf{H} = (0.1,\ 0.1;\ 1,\ 1)\} = 0.05$, $p\{\mathbf{H} = (0.1,\ 1;\ 0.1,\ 0.1)\} = 0.025$, $p\{\mathbf{H} = (0.1,\ 1;\ 0.1,\ 1)\} = 0.1$, $p\{\mathbf{H} = (0.1,\ 1;\ 1,\ 0.1)\} = 0.075$, $p\{\mathbf{H} = (0.1,\ 1;\ 1,\ 1)\} = 0.05$, $p\{\mathbf{H} = (1,\ 0.1;\ 0.1,\ 0.1)\} = 0.025$, $p\{\mathbf{H} = (1,\ 0.1;\ 0.1,$

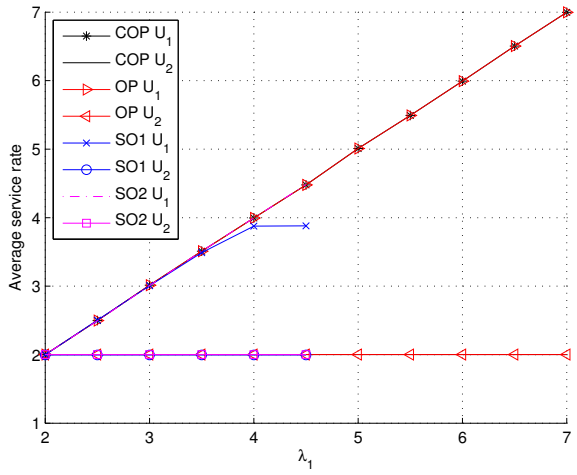

Fig. 29. Average service rates of optimal, close to optimal and suboptimal policies versus λ_1, when $\lambda_2 = 2$.

$1)\} = 0.1$, $p\{\mathbf{H} = (1, 0.1; 1, 0.1)\} = 0.075$, $p\{\mathbf{H} = (1, 0.1; 1, 1)\} = 0.05$, $p\{\mathbf{H} = (1, 1; 0.1, 0.1)\} = 0.025$, $p\{\mathbf{H} = (1, 1; 0.1, 1)\} = 0.1$, $p\{\mathbf{H} = (1, 1; 1, 0.1)\} = 0.075$, $p\{\mathbf{H} = (1, 1; 1, 1)\} = 0.05$.

For a Poisson process, the second moment of arrivals in each frame is finite [12]. Thus, each terminal is assumed to receive packets according to a Poisson process at an average rate of λ_n. The average input rate of terminal 2 (U_2) is fixed to $\lambda_2 = 2$ packets/slot. The maximum transmission power is $P_{\max} = 2$dB. The average input rate of terminal 1 (U_1), λ_1, gets values inside the network stability region illustrated in Fig. 27. The discount factor in (173) is α =0.9 and Lagrangian multiplier in (171) is given as $\beta = [0.6, 0.6]$. The long-term average power, delay, and throughput of each terminal were calculated over $T_0 = 50000$ frames.

In Fig. 28, the average number of active APs for optimal (OP) and close to optimal (COP) control policy is plotted as a function of λ_1. It can be seen in the figure that the long-term average number of active APs for the optimal and close to optimal policies decreases as λ_1 increases. It can also be seen that the average number of active APs is quite close to 1 even for low λ_1. When λ_1 increases, the interference U_1 causes to U_2 in the simultaneous transmission increases. That is why the probability to choose TDMA together with just one AP instead of interference and 2 APs increases, when λ_1 increases. In addition, due to the fixed low arrival rate of U_2 and Gilbert-Elliot channel

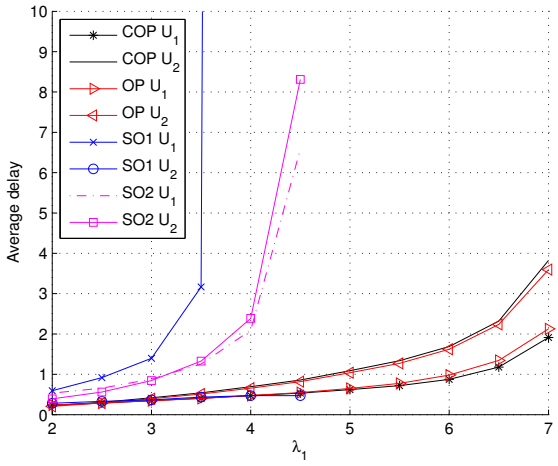

Fig. 30. Average delays of optimal, close to optimal and suboptimal policies versus λ_1, when $\lambda_2 = 2$.

model, it is also quite likely that U_1 uses the whole time slot t and transmits alone to its AP. Especially, when the channel of U_1 is good and the channel of U_2 is bad, it is likely to choose just one active AP due to the interference U_1 would cause to U_2 in the simultaneous transmission.

In Fig. 29, the long-term average service rates of U_1 and U_2 for optimal, close to optimal and sub-optimal policies are plotted as a function of λ_1. Let SO1 and SO2 denote the suboptimal policy 1 and 2, respectively. The results indicate that the performance of the close to optimal policy is very close to the performance of the optimal one. Thus, simplifying the optimization by calculating the value functions separately for each terminal does not have much effect to the performance of a dynamic policy. It can also be seen, that average service rates of the optimal and sub-optimal policies equal λ_n. This supports the stability analysis in Section 5.7, since $(\lambda_2 = 2, \lambda_1 = 7)$ is on the boundary of the network stability region illustrated in Fig. 27. In terms of stability, the maximum supportable arrival rates of the optimal and close to optimal policies are considerably higher than the maximum supportable arrival rate of the sub-optimal policies. Since the resource allocation actions are done separately for each terminal U_n, collisions might happen and the sup-optimal policies cannot support the arrival rates as high as optimal and close to optimal control policies. It can also be seen that the sub-optimal policy 1 is

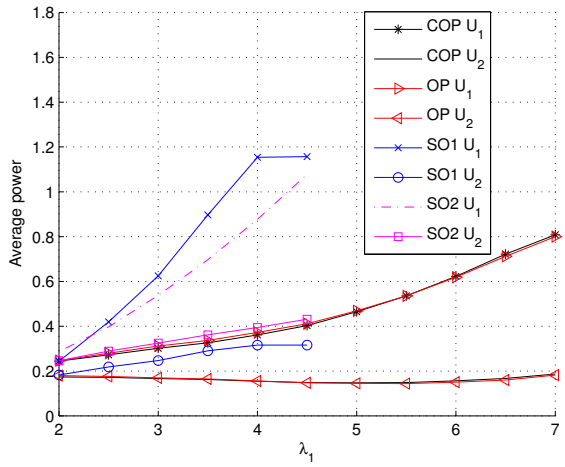

Fig. 31. Average powers of optimal, close to optimal and sub-optimal policies versus λ_1, when $\lambda_2 = 2$.

not stable for arrival rates higher than 3.5, since the average arrival rate is higher than the average service rate when $\lambda_1 > 3.5$.

In Fig. 30, the long-term average delays of U_1 and U_2 for all three policies are plotted as a function of λ_1. The average delay of the close to optimal policy is very close to the average delay of the optimal control policy. However, the average delay of the sub-optimal control policy is significantly longer than the average delays of the optimal and close to optimal policies. This is because the sub-optimal policies do not have full QSI at the terminals.

The long-term average powers of U_1 and U_2 for all polices are plotted as a function of λ_1 in Fig. 31. The average power of the close to optimal policy is very close to the average power of the optimal control policy. Since the sub-optimal actions are chosen separately for each terminal, the long-term average power of the sub-optimal policies increases rapidly with λ_1. It can also be seen in Fig. 31 that the average power of the sub-optimal policy 1 saturates at $\lambda_1 = 4$, where the average transmission power of U_1 equals $P^{\max}$.

5.11 Chapter summary

In this chapter, a new paradigm in wireless network access was presented and analyzed. We considered an advanced wireless technology, where each terminal in an ad hoc or

multi-hop cellular network can work as an AP any time when connected to the Internet. The considered network consisted of L potential APs and a set of N terminals. Dynamic programming tools and VIA are used to derive an optimal control policy.By adapting to the changes in network conditions our policy allocates M out of L APs to terminals in an optimal way and stabilizes the network. Since the optimization protocol is effective only for relative small values of N and L, we introduced the preclustering technique to segmentize the network into smaller subnetworks with relatively small number of candidate APs and user terminals. The interference between these subnetworks was eliminated by using a given frequency reuse factor. In addition, since the optimal control policy requires full knowledge of CSI and QSI, approximate dynamic programming was used to provide two new suboptimal policies, that do not require information on queue lengths. The suboptimal policies were shown to significantly decrease the computational complexity of the problem with controllable loss in performance.

In addition, the K-step Lyapunov drift was used to analyse the stability and the performance of the optimal control policy. We proved the stability of our optimal control policy and showed that the performance of our optimal policy is better than the performance of any other existing network stabilizing strategy.

Finally, the simulation results are provided to support the theoretical analysis of the work presented in this chapter.

The contributions of this chapter can be summarized as follows:

– A novel approach for integrating ad hoc or multihop cellular network and Internet was introduced. This approach enabled a self-organized extension/ shrinking of the network topology and offered maximum capacity in the network nodes with maximum temporal/spatial traffic intensity. The network self-configuration dynamically followed the changes in the temporal distribution of the traffic without additional investments into the network infrastructure.
– A new topology control policy was introduced that a) Minimized the impact of the new established route to the already existing connections in wireless network. b) Calculated the necessary rerouting for all sessions if needed. c) Minimized the overall power consumption in the network, and d) controlled the stability of the network.
– Approximate dynamic programming was used to provide two new sub-optimal control policies.
– We illustrated network stability region and the stability regions of the different control policies.

– We showed that our optimal control policy provides a stable network.

6 Conclusions and future work

In this chapter we first summarize the most important contributions and results presented in this thesis. Then we point out some future research directions in this field.

In this thesis, a number of new paradigms in wireless networks with queues and time varying channels were presented. The contributions included solutions for optimal and sub-optimal dynamic resource allocation, topology control and network stability analysis. We provided unified system models, unified control problems, unified network stability regions and unified stability analysis for both SSP and PSP cognitive networks as well as for CWN. Novel approaches to the optimization and stability analysis of dynamic control policies in wireless networks were also presented in this thesis.

The first chapter considered the motivation behind this research and summarized the previous work on the network stability analysis.

In Chapter 2, we explained the concept of network stability and presented the tools used to analyse the stability of time varying queueing networks.

A unified optimization framework for resource control in computing cloud for both SSP and PSP cognitive networks as well as for CWNs was considered in Chapter 3. As the control problem was to dynamically adjust resources according to changes in channel and workload fluctuations, we formulated the problem into a MDP and used VIA to provide the optimal solution. The resulting dynamic control policy was designed to maximize the long-term average throughput and to minimize the energy cost of the overall system.

In addition, a new unified stability regions and stability analysis for both SSP and PSP cognitive networks and for CWNs were also proposed in Chapter 3. We analysed the K-step Lyapunov drift and proved that contrary to what was proposed in [10], [12], the frame based policy in [12] does not provide performance better than the stationary randomized policy and that the frame based policy does not minimize the bound for the average queue length. We showed that our dynamic control policy stabilizes network and provides performance better than the randomized stationary policy proposed in [12].

The numerical results in Chapter 3 were used to illustrate the performance of the dynamic control policy for cognitive wireless networks, PC networks as well as for CWNs. It was show that by adapting to the changes in network conditions, our dynamic policy mitigates the effect of PSP and SSP cognitive networks on each other. The

simulation results were also used to support the stability analysis presented in this chapter.

In Chapter 4, we addressed a unified cooperative network control problem for SSP and PSP cognitive networks and for CWNs. The goal was to dynamically adapt to the changes in network conditions in order to maximize the long-term average throughput and to minimize the delay. Dynamic programming methods were used to provide a new unified cooperative control policy, that solves the problem for both SSP and PSP cognitive networks and for CWN.

In order to compare the potential performances of the cooperative and non-cooperative communication systems to each other, we illustrated unified cooperative and non-cooperative network stability regions for both SSP and PSP cognitive networks as well as for CWNs. In addition, the concept of InTeNet was proposed in Chapter 4, which provides the maximum capacity in the network.

The K-step Lyapunov drift was used to analyse the performance and the stability of the proposed control policy also in Chapter 4. We showed that when the control actions need to be calculated for each possible network state, the best network stabilizing policy minimizes the maximum queue length over all the terminals. Since in Chapter 4 our optimal control policy is designed to minimize the long-term average maximum queue length over all the terminals, the proposed policy stabilizes the network and provides performance better than other similar dynamic policies.

The simulation results in Chapter 4 were used to support the analytical analysis of this chapter and to compare the performances of cooperative and non-cooperative control policies to each other.

In Chapter 5 a new paradigm in wireless network access was presented and analysed. We considered a dynamic network topology where a certain class of wireless terminals can be turned into and AP any time, when connected to Internet. The goal was to control the network resources and the topology in a way that stabilizes the network, maximizes the long-term average throughput and minimizes the average power consumption at each terminal. Dynamic programming methods and VIA were used to provide an optimal topology control policy that was shown to stabilize the network to minimize the bound for the average queue length.

As the optimal control policy requires full CSI and QSI, we identified the computational complexity of the problem, and used approximate dynamic programming tools to provide two new suboptimal policies. In addition, in order to compare the performances

of the proposed control policies to each other, we illustrated the network stability region, and the stability regions of the optimal and suboptimal policies in Chapter 5.

The numerical results in Chapter 5 were provided to support our stability analysis and to compare the performances of optimal and sub-optimal control policies to each other.

The comprehensive work, analysis and results presented in this thesis open a number of new research directions for future wireless networks. The optimal dynamic control policy can be used as a performance benchmark and to lay foundation for future solutions of different simplified dynamic network stabilizing resource allocation schemes for wireless networks.

The network stability analysis in this thesis will provide additional background for future research directions on this field. As the queuing bound is valid only when the average arrival rates are strictly inside the network stability region, as a future work, it would be interesting consider also a case, where the entire channel capacity could be exploited. In addition, new network stabilizing algorithms that effectively utilize the full channel capacity should be provided in order to minimize the network delay.

References

1. J. Perez-Romero, J. Sanchez-Gonzalez, R. Agusti, B. Lorenzo and S. Glisic, “Power efficient resource allocation in a heterogeneous network with cellular and D2D capabilities", *IEEE Transactions on Vehicular Technology*, vol. , no. , pp. -, Jan. 2016.
2. M. Andrews, K. Kumaran, K. Ramanan, A. Stolyar, and P. Whiting, “Providing quality of service over a shared wireless link", *IEEE Communications Magazine*, 2001.
3. R. Cruz and A. Santhanam, “Optimal routing, link scheduling, and power control in multi-hop wireless networks", in *IEEE Proceedings of INFOCOM*, April 2003.
4. T. ElBatt and A. Ephremides, “Joint scheduling and power control for wireless ad-hoc networks", in *IEEE Proceedings of INFOCOM*, 2002.
5. D. Feng, C. Jiang, G. Lim, L. J. Cimini, G. Feng, and G.Y. Li, “A survey of energy-efficient wireless communications," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 1, pp. 167-178, First, Quart., 2013.
6. H. Li, M. Lott, M. Weckerle, W. Zirwas, and E. Schulz, “Multihop communications in future mobile radio networks," in *Proc. IEEE PIMRC*, Lisboa, Portugal, Sep. 2002, vol. 1, pp. 54-58.
7. B. Coll-Perales and J. Gozalvez, “Energy efficient routing protocols for multi-hop cellular networks," in *Proc. IEEE PIMRC*, Tokyo, Japan, Sep. 2009, pp. 1457-1461.
8. B. Lorenzo and S. Glisic, “Optimal routing and traffic scheduling for multihop cellular networks using genetic algorithm," *IEEE Trans. Mobile Comput.*, vol. 12, no. 11, pp. 2274 - 2288, Nov. 2013.
9. R. Urgaonkar, U. L. Kozat, K. Igarashi and M. J. Neely, “Dynamic resource allocation and power management in virtualized data centers", in *Proceedings of the IEEE Network Operations and Management Symposium (NOMS)*, 2010.
10. M. J. Neely, E. Modiano and C. E. Rohrs, “Dynamic power allocation and routing for time-varying wireless networks", *IEEE J. Sel. Areas Commun.*, vol. 23, no. 1, pp. 89-103, Jan. 2005.
11. B. E. Collins and R. L. Cruz, “Transmission policies for time varying channels with average delay constraints", in *Proc. of Allerton Conf. on Commun., Control, and Comp.*, Monticello, IL, USA, 1999.
12. M. J. Neely, “Dynamic power allocation and routing for satellite and wireless networks with time varying channels", *Ph.D dissertation*, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA, 2003.
13. M. J. Neely, E. Modiano and C. E. Rohrs, “Power allocation and routing in multibeam satellites with time varying channels", *IEEE Trans. on Networking*, vol. 11, no. 1, pp. 138-152, Feb. 2003.
14. L. Georgiadis, M. J. Neely and L. Tassiulas, “Resource Allocation and Cross-Layer Control in Wireless Networks", *Foundations and Trends in Networking*, Now Publisher, 2006.
15. M. Goyal, A. Kumar and V. Sharma, “Optimal cross-layer scheduling of transmissions over a fading multiaccess channel", *IEEE Trans. Inf. Theory*, vol. 54, no. 8, pp. 3518-3537, Aug. 2008.
16. W. B. Powell, “Approximate dynamic programming", *John Wiley and Sons*, Princeton, New Jersey, USA, 2007.

17. R. R. Rao and A. Ephremides, “On the stability of interacting queues in a multiple-access system," *IEEE Transactions on Information Theory*, vol. 34, no. 5, pp. 918-930, 1988.
18. L. Tassiulas and A. Ephremides, “Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," *IEEE Transactions on Automatic Control*, vol. 37, no. 12, pp. 1936-1948, 1992.
19. L. Tassiulas and A. Ephremides, “Dynamic server allocation to parallel queues with randomly varying connectivity," *IEEE Transactions on information theory*, vol. 39, no. 2, pp. 466-478, 1993.
20. L. Tassiulas, “Scheduling and performance limits of networks with constantly changing topology," *IEEE Transactions on Information Theory*, vol. 43, no. 3, pp. 1067-1073, May 1997.
21. M. Kangas, S. Glisic, Y. Fang and P. Li, “Resource harvesting in cognitive wireless computing networks with mobile clouds and virtualized distributed data centers: Performance limits," *IEEE Transactions on Cognitive Communications and Networking*, vol. 1 , no. 3, pp. 318 - 334, Dec. 2015.
22. R. Cogill, B. Shrader and A. Ephremides, “Stable throughput for multicast with random linear coding," *IEEE Transactions on Information Theory*, vol. 57, no. 1, pp. 267-281, Dec. 2010.
23. S. P. Meyn and D. Down, “Stability of generalized Jackson networks", *The Annals of Applied Probability*, vol. 4, no. 1, pp. 124-148, Nov. 1994.
24. J. R. Jackson, “Jobshop-like queueing systems", *Management Sci.*, vol. 10, pp. 131-142, Nov. 1963.
25. S. Kumar and P. R. Kumar, “Performance bounds for queueing networks and scheduling policies," *IEEE Transactions on automatic control*, vol. 40 , no. 8, pp. 1600-1611, Aug. 1994.
26. P. R. Kumar and S. P. Meyn, “Stability of queueing networks and scheduling policies," *IEEE Transactions on automatic control*, vol. 40 , no. 2, pp. 251-260, Feb. 1995.
27. N. McKeown, V. Anatharam and J. Walrand, “Achieving 100% throughput in an input-queue switch," in *Proc. INFOCOM*, pp. 296-302, 1996.
28. E. Leonardi, M. Mellia, F. Neri and M. Marsan, “On the stability of input-queued switches with speed up," *IEEE Transactions on Networking*, vol. 9, no. 1, pp. 104-118, Feb. 2001.
29. S. P. Meyn and R. L Tweedie, “Markov Chains and Stochastic stability," *Springer-Verlag*, First version 1993, updated 2005.
30. M. J. Neely, E. Modiano and C. Li, “Fairness and optimal stochastic control for heterogeneous networks," in *Proc. INFOCOM*, March 2005.
31. M. J. Neely and R. Ungaonkar, “Opportunistic, backpressure, and stochastic optimization with the wireless broadcast advantage", in *Proc. ASILOMAR Conference on Signals, Systems, and Computers*, Pacific Grove, CA, Oct. 2008.
32. M. J. Neely, “Stability and capacity regions for discrete time queueing networks," *Cornell University Library*, Mar. 2010.
33. M. J. Neely, “Queue stability and probability 1 convergence via Lyapunov optimization," *Cornell University Library*, Aug. 2010.
34. M. J. Neely, “Energy optimal control for time varying wireless networks," in *Proc. INFOCOM*, March 2005.
35. Maria Kangas and Savo Glisic, “Throughput Optimal Resource Management of Cooperative Networks with Mobile Clouds," in *PIMRC 2011*, Toronto, Canada, August 2011

36. M. J. Neely, E. Modiano and C. Li, “Fairness and optima stochastic control for heterogeneous networks," *IEEE Transactions on Networking*, vol. 16, no. 2, pp. 296-409, Apr. 2008.
37. M. J. Neely, “Energy optimal control for time varying wireless networks," *IEEE Transactions on Information Theory*, vol. 57, no. 7, pp. 2915-2934, Jul. 2006.
38. E. Yeh and R. Berry, “Throughput optimal control of cooperative relay networks", *IEEE Trans. Inf. Theory*, vol. 53, no. 10, pp. 3827-3833, Oct. 2007.
39. H. Halabian, I.Lambaris and C. Lung", “Network capacity region of multi-queue multi-server queuing system with time varying connectivities", in *ISIT*, 2010.
40. J. Jose, L. Ying and S. Wishwanath", “On the stability region of amplify-and-forward cooperative relay networks", in *ITW*, Oct. 2009.
41. D. Bertsekas, *Dynamic Programming and Optimal Control*, Vol. 1, 3rd ed., Athena Scientific, 2005.
42. D. Bertsekas, *Dynamic Programming and Optimal Control*, Vol. 2, 3rd ed., Athena Scientific, 2007.
43. A. Weiss, “Computing in the clouds," *netWorker*, vol. 11, no. 4, pp. 16-26, 2007.
44. B. Hayes, “Cloud computing," *Commun. ACM*, vol. 51, no. 7, pp. 9-11, 2008.
45. M. Armbrust, A. Fox, R. Griffith and A. D. Joseph, “A view of cloud computing," *Communication of the ACM*, vol. 53, no. 4, pp. 50-58, Apr. 2010.
46. W. Voorsluys, J. Broberg and R. Buyya, “Introduction to cloud computing," *John Wiley and Sons, Inc.*, 2011.
47. *Open Cloud Manifesto.*, Online. Available: https://www.opencloudmanifesto.org/
48. L. M. Vaquero, L. Rodero-Merino, J. Caceres and M. Linder, “A break in the clouds: Towards a cloud definition," *SIGCOMM Comput. Commun. Rev.*, vol. 39, no. 1, pp. 50-55, 2009.
49. F. Chang, J. Dean, S. Ghemawat, W. C. Hsieh, D. A. Wallach, M. Burrows, T. Chandra, A. Fikes and R. E. Gruber ,“Bigtable: A distributed storage system for structured data," *AMC Trans. Computer Systems (TOCS)*, vol. 26, no. 2, pp. 4, 2008.
50. S. Sakr, A. Liu, D. M. Batista and M. Alomari, “A survey of large scale data management approaches in cloud environments," *IEEE Communications Surveys and Tutorials*, vol. 13, no. 3, pp. 311-336, 2011.
51. C. Wang, K. Ren, W. Lou and J. Li, “Toward publicitly auditable secure cloud data storage services," in *IEEE Network*, vol. 24, no. 4, pp. 19-24, 2010.
52. H. T. Dinh, C. Lee, D. Niyato and P. Wang, “A survey of mobile cloud computing: Architecture, applications, and approaches," *Wireless Communications and Mobile Computing (WCMC)*, pp. 1587-1611, 2013.
53. R. Khan, M. Othman, S. A. Madani and S. U. Khan, “A survey of mobile cloud computing application models," *IEEE Communications Surveys and Tutorials*, vol. 16, no. 1, first quarter, 2014.
54. J. Baliga, R. W. A. Ayre, K. Hinton and R. S. Tucker, “Green cloud computing: Balancing energy in processing, storage, and transport," *Proceedings of the IEEE*, vol. 99, no. 1, pp. 149-167 Jan. 2010.
55. M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica and M. Zaharia, “Above the clouds: A Berkeley view of cloud computing," *Electr. Eng. Comput. Sci. Dept. Univ. California, Berkeley, CA*, Tech. Rep., Feb. 2009.

56. X. Fan, J. Cao and H. Mao, “A survey of mobile cloud computing," *ZTE Communications*, vol. 9, no. 1, pp. 4-8, 2011.
57. L. Guan, X. Ke, M. Song and J. Song, “A survey of research on mobile cloud computing," in *Proceedings of the IEEE/ACIS 10th International Conference on Computer and Information Science*, pp. 387-392, 2011.
58. R. Buyya, C. S. Yeo and S. Venugopal, “Market-oriented cloud computing: Vision, hype, and reality for delivering IT services as computing utilities," in *Proc. 10th IEEE Int. Conf. High Performance Comput. Commun.*, China, pp. 5-13, 2008.
59. D. Huang, “Mobile cloud computing," *IEEE COMSOC Multimedia Communications Technical Committee E-letter*, vol. 6, no. 10, pp. 1-7, 2011.
60. D. Kondo, B. Javadi, P. Malecot, F. Capello and D. P. Anderson, “Cost-benefit analysis of cloud computing versus desktop grids," in *Proceedings of the IEEE Int. Symp. Parallel Distrib. Process.*, Italy, May 2009.
61. M. Ali, “Green cloud on the horizon," in *Proceedings of the 1st International Conference on Cloud Computing (CloudCom)*, pp. 451-459, 2009.
62. *Google Docs.*, Online. Available: https://docs.google.com
63. Amazon simple storage service. Online. Available: https://aws.amazon.com/s3/
64. IMB Smart Business Services. Online. Available: https://www.imb.com/imb/cloud
65. A. Klein, C. Mannweiler, J. Schneider and H. D. Schotten, “Access schemes for mobile cloud computing," in *Proceedings of the 11th International Conference on Mobile Data Management*, pp. 387-392, 2010.
66. M. Satyanarayanan, “Fundamental challenges in mobile computing," in *Proceedings of the 5th annual AMC symposium on Principles of distributed computing*, pp. 1-7, 1996.
67. K. Kumar and Y. Lu, “Cloud computing for mobile users: Can offloading computation save energy?," in *IEEE Computer*, vol. 43, no. 4, pp. 51-56, 2010.
68. Khadija Akherfi, Micheal Gerndt and Hamid Harroud, “Mobile cloud computing for computation offloading: Issues and challenges," in *Applied Computing and Informatics*, 2016.
69. M. Gupta and S. Singh, “Greening of the Internet," in *Proceedings of Conf. Appl. Technol. Architectures Protocols Computer Commun.*, Germany, pp. 19-26, 2003.
70. J. Liu, F. Zhao, X. Liu and W. He, “Challenges towards elastic power management in Internet data centers," in *Proceedings of the IEEE Int. Conf. Distrib. Comput. Syst. Workshops*, pp. 65-72, CA, 2009.
71. P. Padala, K-Y. Hou, K. G Shin, X. Zhu, M. Uysal, Z. Wang, S. Singhal and A. Merchant, “Adaptive control of virtualized resources in utility computing environments", in *Proceedings of EuroSys*, 2007.
72. A. Greenberg, J. Hammilton, D. A. Maltz and P. Patel, “The cost of a cloud: Research problems in data center networks", *ACM SIGCOMM Computer Communication Review*, vol. 39, no. 1, Jan. 2009.
73. A. Wolke, M. Bichler and T. Setzer, “Planning vs. dynamic control: Resource allocation in corporate clouds", *IEEE Transactions on Cloud Computing*, no. 99, pp. 1-14, 2015.
74. S. Vijayakumar, Q. Zhu and G. Agrawal, “Dynamic reource provisioning for data streaming applications in a cloud environment," in *Proc.2nd IEEE Int. Conf. on Cloud Comput. Tech. and Science*, pp. 441-448, 2010.
75. L. Liu, H. Wang, X. Liu, X. Jin, W. He, Q. Wang and Y. Chen, “GreenCloud: A new architectures for green data center," in *Proc. ICAC-INDST*, pp. 29-38, 2009.

76. P. Padala, K-Y. Hou, K. G Shin, X. Zhu, M. Uysal, Z. Wang, S. Singhal and A. Merchant, “Automatic control of multiple virtualized resources", in *Proceedings of EuroSys*, 2009.
77. X. Liu, X. Zhu, P. Padala, Z. Wang and S. Singhal, “Optimal multivariate control for differentiated service on a shared hosting platform", in *Proceedings of CDC*, Dec. 2007.
78. B. Li, J. Li, J. Huai, T. Wo, Q. Li and L. Zhong, “EnaCloud: An energy-saving application live placement approach for cloud computing environments", in *IEEE International Conference on Cloud Computing*, Tampa, USA, 2009.
79. D. Kusic and N. Kandasamy, “Power and performance management of virtualized computing environments via lookahead control", in *Proceedings of ICAC*, June 2008.
80. Y. Chen, A. Das, W. Qin, A. Sivasubramaniam, Q. Wang and N. Gautam, “Managing server energy and operational cost in hosting centers", in *Proceedings of SIGMERICS*, June 2005.
81. S. Govindan, J. Choi, B. Urgaongar, A. Sivasubramanian and A. Baldini, “Statistical profiling-based techniques for effective power provisioning in data centers", in *Proceedings of EuroSys*, Apr 2009.
82. W. Xu, X. Zhu and S. Neema, “Online control for self-management in computing systems", in *Proceedings of the IEEE/IFIP Network Operations and Management Symposium (NOMS)*, 2006..
83. C. Yang, Z. Chen, Y. Yao, B. Xia and H. Liu, “Energy efficiency in wireless cooperative caching networks", in *Proc. IEEE ICC*, 2014.
84. R. Kaewpuang, D. Niytao, P. Wang and E. Hossain, “A framework for cooperative resource management in mobile cloud computing", *IEEE J. Sel. Areas Commun.*, vol. 31, no. 12, pp. 2685-2700, Dec. 2013.
85. H. Yue, M. Pan, Y. Fang and Savo Glisic, “Spectrum and energy efficient relay station placement in cognitive radio networks", *IEEE J. Select. Areas Commun.*, vol. 31, no. 5, May 2013.
86. H. S. Wang and N. Moayery, “Finite-state Markov channel - a Useful model for radio communication channels," *IEEE Transactions on Vehicular Technology*, vol. 44, no. 1, pp. 163-171, Feb. 1995.
87. J. M. Park and G. U. Hwang, “Mathematical Modeling of Rayleigh fading channels based on finite state Markov chains," *IEEE COMMUNICATION LETTERS*, vol. 13, pp. 764-766, Oct. 2009.
88. S. Glisic, B. Lorenzo, I. Kovacevic and Y. Fang, “Modeling dynamics of complex wireless networks", in *HPCS*, Helsinki, Finland, July 2013.
89. J. Kephart, H. Chan, R. Das, D. Levine, G. Tesauro, F. Rawson and C. Lefurgy, “Coordinating multiple autonomic managers to achieve specified power-performance tradeoff", in *Proc. International Conf. on Autonomic Computing*, June 2007.
90. D. Kusic and N. Kandasamy", “Control for dynamic resource provisioning in enterprise computing systems", in *Proc. International Conf. on Autonomic Computing*, 2006.
91. E. Kalyvianaki, *Resource provisioning for virtualized server applications*, Computer Laboratory, University of Cambridge, Cambridge, United Kingdom, Tech. Rep., Nov. 2009.
92. R. A. Berry and R. B. Gallager, “Communication over fading channels with delay constraints", *IEEE Trans. Inf. Theory*, vol. 50, no. 1, pp. 125-144, Jan. 2002.
93. D. J. Ma, A. M. Makowski and A. Shwartz, “Estimation and optimal control for constrained Markov chains", in *IEEE Conference on Decision and Control*, 1986.
94. M. Goyal and A. Kumar and V. Sharma, “Power constrained and delay optimal policies for scheduling transmissions over a fading channel", in *IEEE Infocom*, 2003.

95. R. Bellman, *Dynamic Programming*, Princeton University Press, Princeton, NJ, 1957.
96. M. Pan, P. Li, Y. Song, Y. Fang, P. Lin and S. Glisic, “When spectrum meets clouds: optimal session based spectrum trading under spectrum uncertainty," *IEEE Transactions on Mobile Computing*, vol. 32, no. 3, pp. 615-627, March. 2014.
97. D. Tse and P. Viswanath, “Fundamentals of Wireless Communication," *Cambridge University Press*, 2005.
98. A. Nosratinia, T. E. Hunter and A. Hedayat, “Cooperative Communication in Wireless Networks," *IEEE Commun. Mag.*, vol. 42, no. 10, pp. 68-73, Nov. 2004.
99. A. Sendonaris, E. Erkip and B. Aazhang, “User Cooperation Diversity - Part 1: System Description," *IEEE Transactions on Communications*, vol. 51, no. 11, pp. 1927-1938, Nov. 2003.
100. A. Sendonaris, E. Erkip and B. Aazhang, “User cooperation diversity - part 2: Implementation aspects and performance analysis," *IEEE Transactions on Communications*, vol. 51, no. 11, pp. 1939-1948, Nov. 2003.
101. J. N. Laneman, D. N. C. Tse and G. W. Wornell, “Cooperative diversity in wireless networks: Efficient Protocols and Outage Behavior," *IEEE Transactions on Information Theory*, vol. 50, no. 12, pp. 3062-3080, Dec. 2004.
102. G. Kramer, I. Maric and R. D. Yates, “Cooperative Communications," *Foundation and Trends in Networking, Now Publisher*, vol. 1, no. 3-4, pp. 271-425, 2006.
103. R. Urgaonkar and M. J. Neely, “Delay Limited Cooperative Communication with Reliability Constraints in Wireless Networks," *IEEE INFOCOM*, 2009.
104. M. Pan, C. Zhang, P. Li and Y. Fang, “Spectrum Harvesting and Sharing in Multi-Hop CRNs Under Certain Spectrum Supply," *IEEE J. Select. Areas Commun.*, vol. 30, no. 2, pp. 396-378, Feb. 2012.
105. Lumia. Online. Available: http://help.telecom.co.nz/app/answers/detail/a_id/30335/∼/setup-device-as-wireless-modem-for-pc—nokia-lumia-920
106. K. Ur Rahman Khan and R. U. Zaman and A. Venu Gopal Reddy, “Integrating mobile ad hoc networks and the Internet: Challenges and a review of strategies," in *COMSWARE*, 2008.
107. H. Ammari and H. El-Rewini, “Using hybrid selection schemes to support QoS when providing multihop wireless Internet access to mobile ad Hoc networks," *QSHINE*, 2004.
108. I. Koutsopoulos and L. Tassiulas, “Joint optimal access point selection and channel assignment in wireless networks," *IEEE Transactions on Networking*, vol. 15, no. 3, pp. 521-532, Jun. 2007.
109. B. N. Park and W. Lee and C. Lee and C. K. Shin, “QoS-aware adaptive Internet gateway selection in ad hoc wireless Internet access networks," in *Proc. IEEE BROADNETS*, Seoul, Korea, Oct. 2006.
110. W. Li, S. Wang, Y. Cui, X. Cheng, R. Xin, M. A. Al-Rodhaan and A. Al-Dhelaan, “AP association for proportional fairness in multirate WLANs," *IEEE Transactions on Networking*, 2013.
111. R. Kumar, M. Misra and A. K. Sarje, “An efficient Gateway discovery in Ad Hoc Networks for Internet Connectivity," in *IEEE Conference on Computational Intelligence and Multimedia Applications*, 2007.
112. H. Ammari and H. El-Rewiniy, “Integration of mobile ad hoc networks and the Internet using mobile gateways," in *Proc. of the 18th International Parallel and Distributed Processing Symposium*, 2004.

113. C. Huang and H. Lee and Y. Tseng, “A two-tier heterogenius mobile ad hoc network architecture and its load balance routing problem," *AMC Mobile Networks and Applications*, vol. 9, no. 4, pp. 379-391, May. 2004.
114. Y. Tseng and C. Shen and W. Chen, “Mobile IP and ad hoc networks: An integration and implementation experience," *CiteSeerX*, vol. 36, no. 5, pp. 48-55, May. 2003.
115. C.Ahlund and R. Brannstrom and A. Zaslavsky, “Running variance metric for evaluating performance of wireless IP networks in the mobile city testbed," in *Proc. IEEE TRIDENTCOM*, Feb. 2005.
116. F. Hoffmann and D. Medina, “Optimum Internet gateway selection in ad hoc networks," in *Proc. IEEE ICC*, 2009.
117. U. Jonsson and F. Alriksson and T. Larsson and G. Q. Maquire Jr., “MIPMANET-Mobile IP for mobile ad hoc networks," in *Proc. IEEE/AMC Workshop on Mobile and Ad Hoc Networking and Computing*, Boston, MA, USA, Aug. 1999.
118. Y. Sun and E. M. Belding-Royer and C. E. Perkins, “Internet connectivity for ad hoc mobile networks," *International Journal of Wireless Information Networks*, vol. 9, no. 2, pp. 75-78, Apr. 2002.
119. R. Brannstrom and C. Ahlund and A. Zaslavsky, “Maintaining gateway connectivity in multi-hop ad hoc networks," in *Proc. IEEE WLN*, 2009.
120. J. Xi and C. Bettstetter, “Wireless multi-hop internet access: gateway discovery, routing, and addressing," in *Proc. Intern. Conf. on 3G Wireless and Beyond*, San Francisco, USA, 2002.
121. K. U. R. Khan, A. Ahmed, A. V. Reddy and R. U. Zaman, “Hybrid architecture for intregrating mobile ad hoc network and the intrenet using fixed and mobile gateways," *Wireless Days*, 2008.
122. J. Broch and D. Maltz and D. Johnson, “Supporting hierarchy and heterogenious interfaces in multi-hop wireless ad hoc networks," in *Proc. IEEE ISPAN*, Perth, Australia, Jun. 1999.
123. M. Y. Arslan and J. Yoon and K. Sundaresan and S. V. Krishnamurthy and S. Banerjee, “A resource management system for interference mitigation in enterprise OFDMA femtocells," *IEEE Transactions on Networking*, Nov. 2012.
124. D. Lopez-Perez and G. Roche and A. Valcarce and A. Juttner and J. Zhang, ‘Interference avoidance and dynamic frequency planning for WiMAX femtocell networks," in *Proc. IEEE ICCS*, 2008.
125. Y. Sun and R. P. Jover and X. Wang, “Uplink interference mitigation for OFDMA femtocell networks," *IEEE Transactions on Wireless Communications*, vol. 11, no. 2, pp. 614-6258, Feb. 2012.
126. J. Yoo and K. H. Park, “A cooperative clustering protocol for energy saving of mobile devices with WLAN and Bluetooth interfaces," *IEEE Transactions on Mobile Computing*, vol. 10, no. 5, pp. 491-504, Apr. 2011.
127. D. Adelman and A. J. Mersereau, “Relaxations of weakly coupled stochastic dynamic programs," *Operations Research*, vol. 56, no. 3, pp. 712-727, Jan. 2008.
128. M. L. Littman, T. L. Dean and L. Pack Kaelbling", “On the complexity of solving Markov Decision Problems", in *Proc. of the 11th International Conference on Uncertainty in Artificial Intelligence*, 1995.

Book orders:
Granum: Virtual book store
http://granum.uta.fi/granum/

UNIVERSITY OF OULU P.O. Box 8000 FI-90014 UNIVERSITY OF OULU FINLAND

ACTA UNIVERSITATIS OULUENSIS

UNIVERSITY of OULU
OULUN YLIOPISTO

ISBN 978-952-62-1545-7 (Paperback)
ISBN 978-952-62-1546-4 (PDF)
ISSN 0355-3213 (Print)
ISSN 1796-2226 (Online)