

ABSTRACT

KULKARNI, VINEET ASHOK. Social Distance Aware Resource Allocation. (Under the direction of Michael Devetsikiotis.)

Communication flows associated with human end-users will have an underlying social context which determines the importance of the communication. In other words, the network gives us the capability to communicate, but the reason to communicate is external to it. In order to facilitate communication flows, the network has to make decisions on resource allocation to the flows among several others (admission control, traffic policing and so on). If the network remains agnostic to the underlying social context of the communication flow, the resulting decisions will at best be sub-optimal when viewed along the social context dimension.

In this work, we measure the social context associated with a communication flow and incorporate it into resource allocation decisions at the network. We use the notion of social distance between end-users to measure corresponding context. We combine the social distance as declared by the end-user with the overall importance of the user in the social network to derive a social-network-wide social distance measure. Further, we define social distance aware utility functions by imposing maximum achievable utility bounds on the communication flows based on the social distance. This ensures that in an optimal allocation of resources, flows of the same traffic type get differentiated service based on the associated social distance.

We present the resultant resource allocations with respect to wireless networks. Specifically, we look at the case of voice flows competing for resources over an IEEE 802.11e QBSS, and provide theoretical as well as simulation results demonstrating that our social distance aware resource allocation (SDA) achieves higher network utility than IEEE 802.11e for every case considered. We also look at the case of both voice and video calls competing for resources and show that SDA achieves improved network utility as compared to IEEE 802.11e. The reason for this is that SDA allocates resources based on classifying flows through the social distance dimension, as compared to IEEE 802.11e which only takes into account the traffic type for classification.

When users are requesting content hosted on the network, the relationships between content can be used to determine relative importance of content. In the case of social content (Youtube), such relationships are well-defined and thus the social network is already determined. After defining a corresponding social distance measure for videos, we look at three centrality techniques (degree, closeness, betweenness) to determine which of the three performs optimally in determining the most accessed videos. Finally, we look at some of the applications we implemented to determine the feasibility of SDA in a campus environment.

© Copyright 2012 by Vineet Ashok Kulkarni

All Rights Reserved

Social Distance Aware Resource Allocation

by
Vineet Ashok Kulkarni

A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Computer Engineering

Raleigh, North Carolina

2012

APPROVED BY:

Mihail Sichitiu

Wenye Wang

Rudra Dutta

Michael Devetsikiotis
Chair of Advisory Committee

DEDICATION

To my Parents.

BIOGRAPHY

The author completed his B.E. at SDM, Dharwad in the year 2002. After working for 2 years, he did his Masters (M.Tech) at DAIICT, Gandhinagar following which he joined NC State for his PhD.

ACKNOWLEDGEMENTS

I would like to thank my Advisor, Prof. Michael Devetsikiotis for his help in all my work. I am especially grateful for the freedom that he gave us (all of his students) in pursuing out ideas and working on them. I am also grateful to Prof. Mihail Sichitiu for his help and guidance during my PhD. I learnt a lot working on the mesh network project about the hardware of wireless networking for which I am grateful. I want to thank Prof. Wenye Wang and Prof. Rudra Dutta for their encouragement and direction in my work.

I would like to thank Prof. Sichitiu and Prof. Viniotis for trusting me with ECE 470. This course made my life at NC State more enjoyable than almost anything else. I would also like to thank Marhn Fullmer for helping me out of hardware issues innumerable times.

Finally, I would like to thank the students who teach us more than a book ever could.

TABLE OF CONTENTS

List of Tables	vii
List of Figures	viii
Chapter 1 Introduction	1
1.1 Motivation	1
1.2 Objectives of Resource Allocation	3
1.2.1 Flow Allocation and Traffic Matrices	3
1.2.2 Routing and Resource Management	4
1.2.3 Performance Measurement and Network Control	5
1.2.4 Resource Allocation	6
1.2.5 Predicting Traffic Matrix	7
1.3 Implications of Social Context	7
1.3.1 Predicting Traffic Intensity	7
1.3.2 Deriving Social Graphs	8
1.3.3 Other Applications	8
1.4 Scope and Objectives	8
1.4.1 Objectives	9
1.5 Overview of Thesis	10
Chapter 2 Social Networks and Social Distance	12
2.1 Motivation	13
2.2 Social Distance	14
2.2.1 Centrality and Social Distance	15
2.3 Utility Optimization	16
2.3.1 The function f	20
2.4 Social Distance in Content	21
2.4.1 Rank Distribution	21
2.4.2 Content Caching and Distribution	22
2.5 Summary	22
Chapter 3 Voice Call Capacity Analysis	24
3.1 Background	25
3.2 Simulation Setup	25
3.3 Call Capacity Models	26
3.3.1 Single Hop Wireless Network	27
3.3.2 Two Hop Wireless Network	27
3.3.3 Three Hop Wireless Network	28
3.4 Sensitivity Analysis of the Call Capacity	30
3.5 Summary	30

Chapter 4 Resource Allocation for Voice Calls	31
4.1 Overview	31
4.2 IEEE 802.11e QBSS and SDA Theoretical Analysis	32
4.2.1 Utility Maximization	33
4.2.2 Comparing IEEE 802.11e and SDA	36
4.3 IEEE 802.11e and SDA Simulation	39
4.3.1 Simulation Setup	39
4.3.2 Utility in terms of AIFS	40
4.3.3 Social Graph	42
4.3.4 Comparing IEEE 802.11e and SDA	43
4.4 Summary	43
Chapter 5 Resource Allocation for Voice and Video Flows	44
5.1 Using Social Distance for Voice and Video Traffic	44
5.1.1 Simulation Setup	45
5.1.2 Voice and Video Traffic Performance	47
5.1.3 Discussion of Detailed Results	49
5.1.4 Resource Allocation Patterns	50
5.2 Social Distance and Fairness	51
5.3 Summary	52
Chapter 6 Social Distance Aware Content Distribution	53
6.1 Motivation	54
6.2 Data Collection	55
6.3 Timescales	56
6.4 Centrality	57
6.4.1 Degree Centrality	58
6.4.2 Closeness Centrality	58
6.4.3 Betweenness Centrality	59
6.5 Performance Analysis	59
6.5.1 Standard Feeds	60
6.5.2 Science and Technology Category Trace	61
6.5.3 Number of videos served	61
6.6 Proposed L2 Distributed Content Cache	62
6.7 Summary	63
Chapter 7 Experimental Data Trace Collection for SDA	64
7.1 WiFi Localization	64
7.1.1 Generation of the wireless map	65
7.1.2 Localization of a user	66
7.2 Client Applications	67
Chapter 8 Conclusion	69
References	71

LIST OF TABLES

Table 4.1	Network performance comparison for IEEE 802.11e vs SDA when voice calls are competing with best-effort traffic.	38
Table 4.2	VoIP traffic R -value, MOS, and distance modified MOS	39
Table 5.1	IEEE 802.11e vs SDA performance comparison for a traffic mix consisting of both voice and video flows.	47

LIST OF FIGURES

Figure 1.1	Current service differentiation schemes cannot classify voice (or video) flows into further priority classes. In Figure 1.1a, a high priority voice call is blocked due to aggressive low priority voice calls. In Figure 1.1b, a high priority video flow is blocked, when in the same scenario, a high priority voice call would have been admitted.	2
Figure 1.2	The scope of our work within the broader area of Traffic Engineering. . .	4
Figure 1.3	Routing algorithms compute shortest paths based on link costs. Flow allocation depends on shortest paths to determine the optimal path and egress router for each flow. The traffic matrix depends on the flow allocation, and changes the link cost thereby requiring routing updates. . . .	5
Figure 1.4	The scope of our work within the broader area of Traffic Engineering. . .	9
Figure 2.1	An example organizational social network. Each user identifies his/her peers and the perceived priority of the corresponding relationship to him/her personally.	13
Figure 2.2	Prioritized resource allocation among three flows of the same traffic type.	19
Figure 2.3	Rank distributions for two categories of Youtube videos.	22
Figure 3.1	The simple chain topology used for our simulations.	25
Figure 3.2	System Response for 2 Hops (Figure 3.2a) and 3 Hops (Figure 3.2b) are shown. For both cases, one of the links is set to have the data rate of 11Mbps, while the other data rates are varied.	28
Figure 3.3	Partial derivatives of the call capacity (for 3 hops) with respect to the packetization interval.	29
Figure 4.1	The network is made aware of the social distance dependent utility, due to which it can allocate resources differently to flows of the same traffic type. Resource allocation is achieved by setting parameters on the client stations as well as the AP in the case of a wireless network.	32
Figure 4.2	Comparison of the total network utility achieved for IEEE 802.11e vs Social Distance Aware Resource Allocation (SDA). SDA outperforms IEEE 802.11e for every case.	36
Figure 4.3	Maximum Call Capacity, 802.11e vs Distance-aware. 802.11e VoIP call capacity saturates at 4 calls, hence it can accommodate no calls at distances of 2 and 3 respectively. Total network utility for VoIP calls, plain 802.11e vs distance-aware. As it can be seen, the network can almost double its utility to the users.	42

Figure 5.1	The network topology used for simulation is shown in 5.1a. Nodes are placed around a circle of radius 100m with the AP at the centre. Transmission range of all the nodes is set to 250m. 5.1b shows the modification of IEEE 802.11e AC's to include social distance aware traffic AC's. The corresponding AC parameters are also listed.	46
Figure 5.2	Network Performance for a mix of voice and video calls competing for the shared channel. Results for NV=NVI, NV=2NVI and 2NV=NVI are shown in Figure 5.2a, Figure 5.2b, Figure 5.3a respectively. SDA performs better during saturation traffic conditions as compared to IEEE 802.11e. .	48
Figure 5.3	Specific resource allocation patterns for 5.3a are shown in 5.3b for IEEE 802.11e, and in 5.3c for SDA. White cells represent resources were allocated, black cells represent no resources were allocated, and grey cells represent no resources were requested. It can be seen that IEEE 802.11e allocates all requested resources to voice while starving video flows beyond 8 flows. SDA allocates resources to both voice and video flows in χ^1 before allocating resources to voice and video flows in χ^2 and so on. . . .	50
Figure 6.1	Structurally Figure 6.1b has more impact on design and performance decisions than Figure 6.1a, but both have the same ex-or counts. Variations in the inter-relationships are plotted against time for Top Rated (Figure 6.1c), Most Viewed (Figure 6.1d) and Most Recent (Figure 6.1e) videos.	56
Figure 6.2	Comparison of the three centrality methods for the Top Rated videos feed. Betweenness centrality saturates earlier than the other two, it also performs poorly at choosing the best videos as compared to closeness centrality. Also, at small content cache sizes, closeness centrality performs exceptionally well than the other two.	59
Figure 6.3	Figure 6.3a shows the number of hits for cached videos ranked by closeness centrality. Each cell (square) represents 16 videos, with the lower-left cell representing the most viewed and the upper-right representing the least viewed videos. Figure 6.3b shows that betweenness centrality performs poorly in choosing the right videos, since the number of hits is low across the board.	61
Figure 6.4	The performance of cache for Science and Technology trace is shown in Figure 6.4a. Figure 6.4b shows a proposed L2 distributed content cache. .	62
Figure 7.1	Creating a wireless signal map of the building requires capturing RSSI information at different locations in the building.	65
Figure 7.2	Sequence of messages exchanged to achieve the desired objective as determined by the itinerary server.	67

Chapter 1

Introduction

Communication flows over the network are initiated by human end-users for broadly one of two reasons: the need to communicate with another end-user through a real-time communication, or the need to view content hosted on the network which then necessitates transfer of content. The objective of the network is to achieve the desired level of service (QoS) for end-users through the available resources. The service differentiation among such competing flows is dependent on the media of communication (voice, video, text) and also on the importance of the flow to the end-users themselves. Not all voice flows competing for resources are equally important, and a service differentiation scheme which prioritizes flows solely based on traffic type will result in a sub-optimal allocation of resources. We look at the broader traffic engineering problem, and the role of resource allocation within this in determining network performance.

1.1 Motivation

Resource allocation algorithms have been traditionally designed to optimally allocate available resources among the competing flows while satisfying certain capacity and routing constraints imposed by the design of the computer network. In doing this, the algorithms are agnostic to the underlying real-world *social context* associated with the flow that is requesting resources. This is in agreement with a core philosophy of the Internet - that of preserving anonymity of flows. In order to accommodate inelastic flows, a degree of service differentiation is eventually enforced through classification of the flows based on traffic type. Since different inelastic traffic types (voice, video streams) have differing expectations of the QoS (defined in terms of delay, packet loss ratio, data rate), such a classification makes sense.

Networks have finite resources, and the allocation of these resources is FCFS (First Come First Serve), unless the network has the capability to pre-empt flows. We do not consider pre-emption in this work, and assume that the network tries to serve all the flows which have

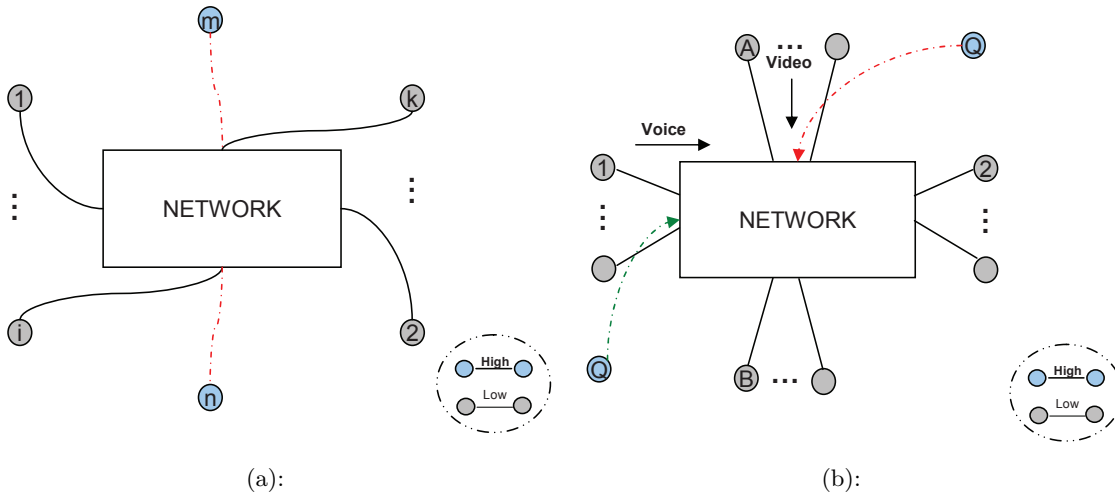


Figure 1.1: Current service differentiation schemes cannot classify voice (or video) flows into further priority classes. In Figure 1.1a, a high priority voice call is blocked due to aggressive low priority voice calls. In Figure 1.1b, a high priority video flow is blocked, when in the same scenario, a high priority voice call would have been admitted.

been accepted. Service differentiation schemes help prioritize resources between accepted flows, differentiating based on the traffic type (inelastic/elastic, voice/video). Service differentiation alone can only provide *graded* service to the accepted flows. It cannot provide *guarantees* on the achieved QoS of a given inelastic flow because no resources are reserved beforehand. Call admission control schemes can be used to ensure that all accepted flows are allocated the minimum necessary resources, and none of the accepted flows are dropped due to scarcity.

In such a scenario, once the capacity of the network is attained, no more flows can be admitted until at least one flow leaves the system. This means that the instance in time at which a flow requests resources determines whether it is admitted into the network or not. More specifically, the network state at the time instant of the incoming resource request determines the outcome. Since the network is agnostic to the underlying social context of the flows, a high priority flow may be denied access when low priority flows are using the network (shown in Figure 1.1). Note that the low priority flows may also all be inelastic (e.g., voice) flows. When all the flows competing for resources belong to the same traffic type, e.g., voice (Figure 1.1a) a high priority voice call may get blocked due to low priority calls which are aggressive in requesting resources. The request rate (for resources) determines allocation rather than the perceived priority. Thus, knowledge of the social context can help identify the high priority flows (even among flows of the same traffic type) and can be used to improve service differentiation.

Classification by traffic type also treats some inelastic flows unfairly. In stating this, we are

assuming that elastic flows do not have any minimum quality requirement from the network, and use up whatever available resources there are. When there is an explicit priority for one type of inelastic traffic over another (voice is higher priority than video), voice calls can effectively starve video flows out of resources as shown in Figure 1.1b. Since the video flows have a minimum quality requirement, the flows begin to fail. When viewed along the social context dimension, the voice flows may actually be lower priority than the video flow requesting resources. A service differentiation scheme which prioritizes flows solely based on the traffic type cannot serve the high priority video flow at the expense of low priority voice calls. The implicit prioritization in these schemes is that voice traffic is always more important than video. Again, social context can be used in such a scenario to elicit the real-world priority of flows to provide improved service differentiation.

In situations where network resources are scarce, anonymity of flows (no knowledge of the social context) can in fact result in suboptimal and unfair allocations of resources. Thus, the social context needs to be determined and communicated to the network in order to achieve optimal allocation of resources. Current service differentiation schemes classify flows by simply looking at the traffic type dimension. Social context can be viewed as a different dimension for ascertaining priority of competing flows. Incorporating both these dimensions into resource allocation algorithms will result in a much more fine-grained (and fairer) service differentiation among flows.

1.2 Objectives of Resource Allocation

Resource allocation algorithms are categorized as part of broader Traffic Engineering functions of the network [1]. The goal of Traffic Engineering is to provide an improved experience to end users of the network with the available resources. The various functions of Traffic Engineering include measuring network performance, identifying and alleviating bottlenecks, controlling resource allocation, dynamically updating flow routing to improve reliability of the network among others. Several of these functions are coupled, and thus a change in one of the functionalities affects other levels of Traffic Engineering. For example, a change in the determination of routes in the network will affect flow allocation decisions and hence bottlenecks in the network. In this section, we look at the objectives of resource allocation algorithms with respect to the broader Traffic Engineering problem.

1.2.1 Flow Allocation and Traffic Matrices

Flows requesting resources from the network are allocated paths based on routing decisions, when the resources are available. Traffic matrices are measurements of resource utilization over

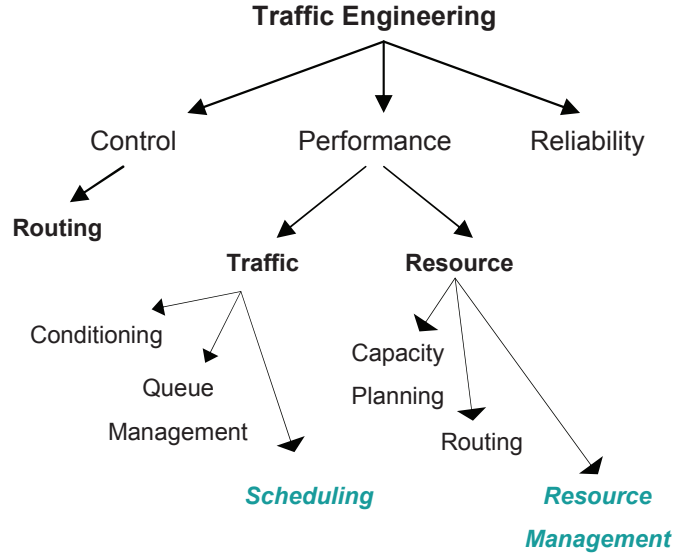


Figure 1.2: The scope of our work within the broader area of Traffic Engineering.

time due to the flow allocation decisions [2][3]. They indicate the traffic flow from the ingress (point at which flow enters the network) to the egress (point at which flow leaves the network). This can then be used to improve route computation by updating link costs according to their utilization.

There have been studies which correlate social networks (and social context) to the resultant traffic demands from the network [4]. In other words, the relationships between users are used to predict frequency of communication between the users. Traffic matrices may then be predicted (instead of measured explicitly) and the network can pro-actively control and engineer the flows to improve performance rather than react to changes based on usage measurements. In this work, we do not study the traffic profiles (temporal behavior) generated due to social network relationships. Our focus is towards identification of the social context to prioritize competing flows for a given network state. We also claim that the relationships of a user change at a much higher timescale than the duration of a single flow. Thus, the social context associated with a flow between two given users is *static* for duration of the flow. The objective of social distance aware resource allocation is thus to provide a higher access probability depending on priority, and not to minimize traffic bottlenecks.

1.2.2 Routing and Resource Management

Traffic matrices are used to update link costs so as to reflect bottlenecks in the network. Routing algorithms then update the routes through the network to alleviate these bottlenecks. We do

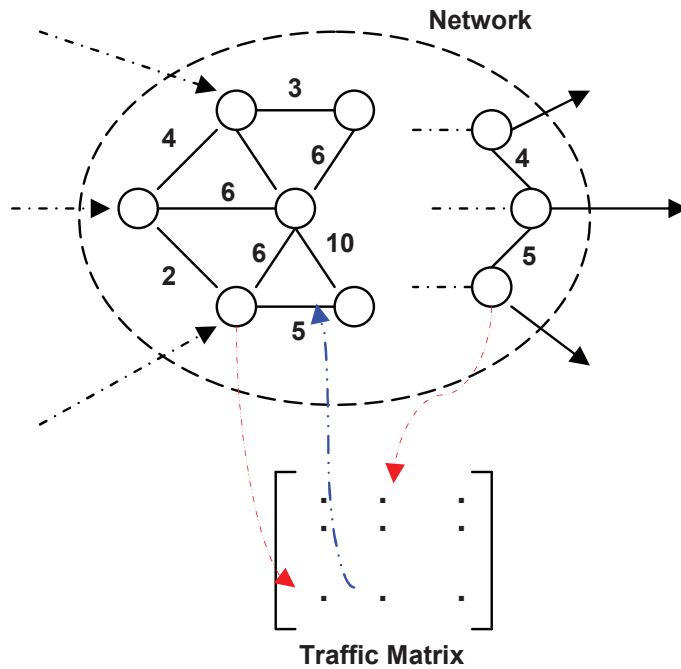


Figure 1.3: Routing algorithms compute shortest paths based on link costs. Flow allocation depends on shortest paths to determine the optimal path and egress router for each flow. The traffic matrix depends on the flow allocation, and changes the link cost thereby requiring routing updates.

not go into the details of timescales at which routing updates happen and their resultant effect on flow allocation. Both routing and flow allocation are thus coupled functions, and a change in routes will change the traffic matrix and vice versa. Routing decisions affect resource allocation decisions too. If the available capacity in the network is not reflected in the routes (route update timescales) flows will be blocked due to congestion in the shortest paths. As we can see, the problem of optimal operation of a network is highly coupled among several subproblems of Traffic Engineering. In this work, we look at the problem of allocating *available* resources among competing flows. We do not consider the problem of identifying unused resources in the network, or the timescales at which such resources are reflected in available routes.

1.2.3 Performance Measurement and Network Control

Traffic measurements can be used to police flows, and prevent the network from becoming congested. Decisions at queue management, as well as at the scheduling component of routers will impact end user experience of the network. We do not look at the problem of determining

traffic profiles depending on end user applications as well as the social context. The focus of our work is categorization of competing flows at a given network state based on their social context, and subsequent differentiation in allocated resources based on the social context. As such, there are no *guarantees* on the achieved quality of service of a given flow. In order to provide service differentiation some amount of control is exerted over the network parameters. But such control is based on the *static* social context (of the external social network) given the state of the network, rather than controlling the network state itself. Our resource allocation algorithm augments traditional resource allocation with knowledge of the social context, and works along side traffic regulation policies to maintain a functional network. We do not change or replace the aspects of Traffic Engineering which deal with traffic regulation.

1.2.4 Resource Allocation

The traffic engineering problem encompasses several subproblems, which are highly coupled together in determining the optimal operational conditions for a network. Resource allocation is one part of the broader traffic engineering objectives. Resource allocation functions are augmented as part of the flow allocation problem, while resource management functions are part of the routing problem. As part of flow allocation, resource allocation algorithms are responsible for implementation of the following competing goals.

Fairness in allocation

When n number of flows are requesting access to network resources, the resource allocation algorithm should be able to accommodate the most broad-based set of flows. This means that a single (or a set of) flow(s) should not be allowed to starve the other flows for resources just by virtue of being more aggressive.

Priority based differentiation

A competing goal to the previous one, is to be able to provide a level of service differentiation between flows depending on their differing expectations of QoS. This is necessary to accommodate real-time flows in the network competing alongside elastic flows for network resources. The real-time flows have a strict QoS requirement that needs to be met in order for the flow to be a useful communication.

Resource reservation

When the network is capable of reserving resources beforehand, the resource allocation algorithm should facilitate reservations for flows. Resources are reserved a-priori for flows which

demand *guaranteed* service from the network. Priority based differentiation can only provide a relative differentiation in access to resources.

1.2.5 Predicting Traffic Matrix

Due to the feedback-based control design of traffic engineering solutions, the decisions of resource allocation algorithms affect other problems which were discussed earlier in this section. Resource allocation directly influences the resulting traffic matrices, though predicting the changes in traffic matrix due to resource allocation is difficult. This is due to the variation in traffic profiles of individual flows over time, which may appear as a distinct characteristic in the multiplexed flow stream too. Thus, we only focus on the resource allocation algorithm goals of providing priority based differentiation between flows while ensuring fairness, and do not concern ourself in this work with predicting the resulting traffic matrices.

1.3 Implications of Social Context

There may not be an interaction of a human end user which could be classified as truly random, and completely devoid of any social context. The network gives us the capability to communicate, but the reason to communicate is external to it. While the end users can work with an abstraction of the network to achieve their objectives, the network would only function sub-optimally if it too works on an abstraction of incoming traffic (as all flows being equally important). Thus, a knowledge of the social context is important to classify flows based on their real-world priorities. Social context can impact more facets of traffic flows than simply identifying the relative priority. In this section, we look at some of the proposed applications of social distance (and its implications) in optimizing network performance.

1.3.1 Predicting Traffic Intensity

Perceived social distance in social networks between relationship peers has been used to predict the resulting traffic intensity between the associated peers. Specifically, communication flow intensity is predicted based on social relationship weights between users [4]. Call arrival rates in telephone networks are predicted based on social closeness of end users [5]. Understanding the instinctive properties of human communication is important to predict quantities such as the frequency of communication, time instants at which requests for communication usually arrive and so on. We do not focus on determining the existence of a correlation between relationship weights and the corresponding traffic flow properties. Our focus in this work is on classification of flows and not on prediction of their arrival times. Given a set of competing flows, we intend to use the social context to differentiate between them.

1.3.2 Deriving Social Graphs

The converse of the previous problem is to monitor traffic intensity between individuals on the network and *predict* the underlying social network which generates this traffic profile [6]. This is attractive in situations where explicit identification of users and their relationships is not feasible or desirable. In such a scenario, a correlation of traffic intensity to individuals can generate a (anonymized) social network with social distances, which can then be used to classify, profile and police resulting flows. We do not know of any studies which prove that the traffic-flow based (derived) social network is identical/ β -approximate to the actual social network between the end users.

While every interaction of a human involves some social context, the influence of such social context on *frequency* of communication is not distinct. The parameters which completely determine flows in both social and temporal dimensions comprise more quantities than the social distance itself. In this work, we do not focus on predicting relationships based on traffic intensity between users.

1.3.3 Other Applications

Social network relationships are used to study the behavioral patterns of humans [7], such as the organizational patterns based on one or more distinguishing properties (age, profession, religion and so on). This is more the focus of social engineering and sociology fields of study. In terms of computer network performance, social distance has been used to identify spam [8], improve search [9], determine optimal next hops in delay tolerant networks [10] and so on.

1.4 Scope and Objectives

The scope of this work is limited to applying knowledge of social context between end users to resource allocation decisions in the underlying network in order to improve the network performance (Figure 1.4). The social context is not derived based on any traffic monitoring between end users of the network. In our work, social context is in fact represented through the notion of social distance between users. The social distance is just a measure of the perceived relative priorities of relationships (of a user) to him/her personally. We do not derive it from traffic intensity measurements, it is declared a-priori by the users themselves through choosing values from a common reference scale of social distance. In our work related to studying social relationships between content, social distance is correspondingly measured based on the relative popularity of content.

The network which we consider for our work is a wireless network, which works according to the IEEE 802.11 MAC standard. When considering service differentiation for real-time flows,

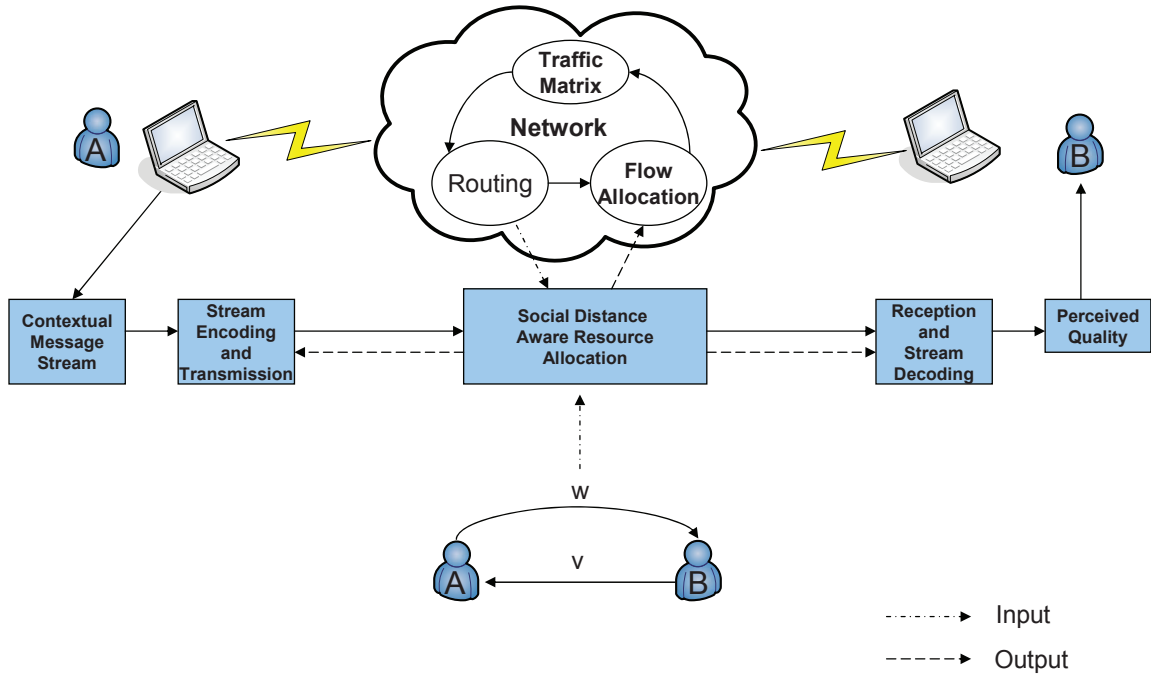


Figure 1.4: The scope of our work within the broader area of Traffic Engineering.

we consider the specific case of a IEEE 802.11e QBSS forming the underlying wireless network. We focus on the case of a single collision domain with all communication happening through an access point (AP).

1.4.1 Objectives

The objectives of our work are as follows:

Service differentiation by social distance

Flows should be differentiated based on their associated social distance, in addition to their traffic type. Traditional service differentiation algorithms classify flows solely based on the traffic type. By incorporating social distance into priority determination we intend to show that even flows belonging to the same traffic type are serviced differently by the network.

Increase network performance

By allocating resources based on the added dimension of social distance (in addition to traffic type), we intend to show that the network performance is improved for the same available resources. The end user experience is in line with their expectations which is determined based on the associated social distance.

Real-time flows differentiation

In an IEEE 802.11e QBSS, there is an explicit priority for voice traffic over video. We intend to classify all real-time flows based on the social distance associated with the flows instead of the traffic type, thereby ensuring a fair access to all real-time flows to the network.

Content distribution by social distance

Socially related content can be a good predictor of access patterns of end users. For the specific case when the content is invariant (e.g., video), caching decisions are relevant and can improve the end user experience along with network performance. We look at the problem of determining candidate content items for caching and distribution over the network.

1.5 Overview of Thesis

The thesis is organized as follows. In Chapter 2 we define the social distance associated with end-users, and propose a way to measure the social distance for all users in the network. This local measure of social distance is augmented with the global centrality measures of the end-users to derive a social-network-wide social distance measure. We define the generic form of a social distance aware utility function, and provide examples for the case of elastic traffic.

In Chapter 3 we look at the problem of determining a closed form expression for voice call capacity in wireless networks. We do this in order to derive some notional bounds for the maximum acceptable number of calls in a wireless network. Following this, in Chapter 4 we compare the performance of SDA with IEEE 802.11e QBSS for the case of voice calls. We provide theoretical as well as simulation results which demonstrate that SDA achieves increased network utility as compared to IEEE 802.11e for every case considered. In Chapter 5 we look at the case where voice and video calls are competing for the shared channel. We show that SDA allocates resources based on the social distance rather than solely based on traffic type (as is the case for IEEE 802.11e).

In Chapter 6, we look at the case where end users are accessing content hosted on the network, and there are inter-relationships between the content items. In such a scenario, the importance of individual items can determine the future user access patterns. We investigate

this for the case of three centrality measures (degree, closeness, betweenness), which are used to compute the relative importance for individual content items and cache them. Through a future access list, we determine which of the three centrality measures provide the best cache performance. Finally, in Chapter 7 we look at some of the applications that we implemented in order to demonstrate the feasibility of SDA in a campus environment.

Chapter 2

Social Networks and Social Distance

Users communicating through real-time flows over the network initiate those flows due to an underlying reason, a *context* associated with the flow. The arrival of such flows into the network is thus driven by the social interrelationships of a user, and the level of importance that he/she attaches to them. The expected level of quality is also different for incoming flows depending on the social context. Consider, for example, an organizational network where the users are categorized as managers and employees. Suppose that the users of this social network are communicating with each other over a shared network using voice calls. When two concurrent voice calls are competing for resources, the social context can provide a good measure of relative priority between the voice calls. For example, a call between two managers should be treated as more important than a call between two employees. When several such users are competing for resources over a shared network, eliciting the underlying social context enables the network to differentiate flows based on their real-world priorities irrespective of their traffic type.

In this chapter, we first provide a definition of social distance as used in this work. Social distance can be thought of as the edge weight (of a relationship) in the social graph. The social graph is determined by users declaring their relationship peers along with the perceived importance of these relationships (social distance) to them *personally*. All users identify the importance of their relationships using the same reference scale. This is done in order to keep the notion of social distance simple. Through this information, we then generate a social-network-wide social-distance measure, which combines the user's personal preferences with their (the user's) overall importance in the bigger social network. This is necessary to bring out the true relative priority of the relationships when looked at from the social network perspective. We illustrate this using a simple example of a social network. In order to incorporate social distance into resource allocation algorithms, we derive the generic form of a social distance aware utility function. We illustrate the effect of such a utility function on resource allocation for the case of elastic flows in the network.

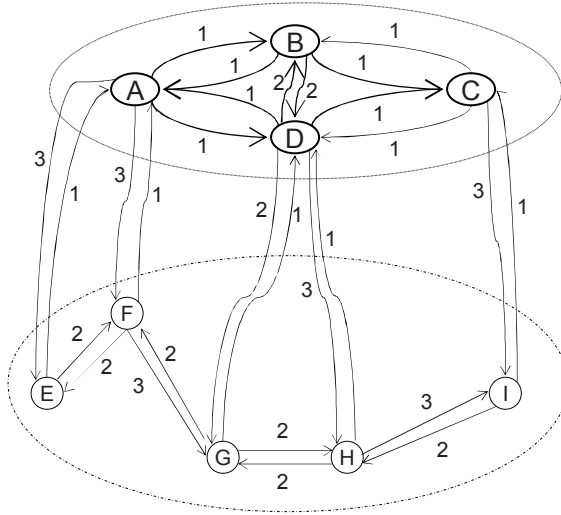


Figure 2.1: An example organizational social network. Each user identifies his/her peers and the perceived priority of the corresponding relationship to him/her personally.

2.1 Motivation

We define the context associated with a flow to be the social distance between the users initiating that flow. This is motivated by the fact that most of our interactions through the network are guided by our social interrelationships and their perceived importance. Knowledge of the social distance can aid the network in categorizing flows according to their real-world priority, rather than solely based on the traffic type. At the macro level, traffic type based differentiation still works for distinguishing real-time flows from non real-time flows. But at a microscopic level, such as looking at just the real-time flows in the network, we believe that social distance provides a more reliable measure of priority determination as compared to the traffic type.

Social distance, as defined and used in Sociology, is variously used to measure both qualitative [11][12] (degree of closeness, feeling of sympathy) as well as quantitative (frequency of interaction between social groups) information about an underlying social network. For our work, social distance is used to measure the relative priority or importance of a relationship to a user. Each user identifies his/her relationships with other users in the network. The user also provides a measure of the *importance* of this relationship to him/her relative to all of their remaining relationships. This does not mean that for a high priority relationship, the frequency of communication with this peer will always be high [4][13]. The social distance is used only to distinguish between competing flows when the resources are limited and prioritization is necessary. In such a scenario, the real-world priority of the flow is the best indicator of desired

service rather than the traffic type. Thus, social distance is used as an indicator to the network about the real-world priority associated with the flow.

Social distance can be used to optimize current communication protocols by making the network *aware* of this implicit relative priority. In the next two sections we look at the case where human users are communicating through the network, and present an example of how social distance can be used to improve resource allocation in such a scenario.

2.2 Social Distance

Communication between any two (human) end users of a network will, in most situations, be determined first by the existence of a social relationship between them, and second by its weight in the bigger social network. The existence of a social relationship can be ascertained by an explicit acknowledgement by the end users themselves. Each user a-priori declares all of his/her peers in the social network. The mere existence of a relationship does not indicate frequency of communication. This is also why inferring a social network by observing traffic on the network may not provide us with the real-world priorities of social relationships. Such an activity would provide us with information about the traffic profile and hotspots, and aid in load balancing in the network. However, our goal is to differentiate between flows based on their real-world priorities.

Along with the social relationships, end users will provide a measure of the perceived importance of the relationship to *them, personally*. Since the perception may indeed be different for peers of the same relationship (employee-manager), the resulting social graph is asymmetric. Each user will choose the social distance (importance) of the relationship from a common reference scale. Since the user indicates *distance* of a peer to him/her, a smaller social distance denotes higher importance, and thus higher real-world priority. Thus, we need a common reference scale with a notion of closeness to define the social graph.

Definition Given a community overlay network S , there exists a real number scale ζ together with a notion of closeness (or proximity) which can be used to define relationships. We can identify the relationship between a pair of communicating entities i and j in this community overlay network by choosing a representative element $\chi_{ij} \in \zeta$. We call this representative element χ_{ij} the *social distance* between entities i and j .

The social distance scale ζ is common for all the users, and so is the definition of closeness. In Figure 2.1 for example, the social distance scale is the set $\zeta = \{1, 2, 3\}$, where 1 is the closest (highest priority) social relationship and 3 is the farthest (lowest priority). From this information, we can distinguish between the relationships of a single user. However, when two different sets of peers are competing for network resources, we do not have a global measure

to differentiate between such flows. The priority of the relationship to the peers has been ascertained, but its importance in the wider social network has still not been measured. We achieve this by including the centrality of the end users in the definition of the social distance.

2.2.1 Centrality and Social Distance

As the social network is usually represented as a graph (Figure 2.1), where the nodes are the users and the edges represent relationships, centrality measures can be defined for nodes in the social network. In fact, social network analysis uses the centrality of a node to quantify its relative importance in the social network, with a node having higher centrality perceived as being more important in the social network. Social networks are also known as scale free networks, because a few nodes will have a high number of connections (called hubs). A centrality measure helps identify such nodes, and hence the scale free nature of the graph itself.

There are several centrality measures defined in literature, such as degree, closeness, betweenness [14][15][16][17]. We use one such measure called the eigenvector centrality. This measure was proposed by Bonacich [18], and it is based on the idea that the eigenvector corresponding to largest eigenvalue of the adjacency matrix (of the social network) is a good measure of the relative weights of nodes in the network. Eigenvector centrality measures the importance of a node in terms of the centrality of its neighbors.

We compute the importance of nodes in Figure 2.1 by evaluating eigenvector centrality based on the chosen relationship weights. This results in the hubs getting a higher centrality measure, and delineating them from the rest of the users. The eigenvector centrality is computed by first determining all the eigenvalues for the adjacency matrix of Figure 2.1. The largest eigenvalue is chosen, and the eigenvector corresponding to this eigenvalue is computed. This eigenvector represents the relative weights of nodes in the social network.

For a given node, we now have two measures which we need to combine. First, the self assessed (in isolation by the node) relationship distances, and second the eigenvector centrality of the node. Continuing with the convention of a smaller social distance representing higher priority we define the network-wide social distance to be:

$$\chi_{ij}^I = \left(\frac{\chi_{ij}^i}{C_i} \right) \quad (2.1)$$

where, χ_{ij}^i is the self-assessed relationship weight by node i , C_i is the eigenvector centrality of node i and χ_{ij}^I is the corresponding global social distance measure for node i . We omit the superscript on χ_{ij} , and it is in fact χ_{ij}^I which is being compared between competing flows. The modified social distance matrix looks as follows:

$$\chi = \begin{bmatrix} 0 & 3.4 & 0 & 3.4 & 10.3 & 10.3 & 0 & 0 & 0 \\ 1.6 & 0 & 3.4 & 6.8 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3.4 & 0 & 3.4 & 0 & 0 & 0 & 0 & 10.3 \\ 1.0 & 6.8 & 3.4 & 0 & 0 & 0 & 6.8 & 10.3 & 0 \\ 2.5 & 0 & 0 & 0 & 0 & 6.8 & 0 & 0 & 0 \\ 1.5 & 0 & 0 & 0 & 6.8 & 0 & 10.3 & 0 & 0 \\ 0 & 0 & 0 & 3.4 & 0 & 6.8 & 0 & 6.8 & 0 \\ 0 & 0 & 0 & 3.4 & 0 & 0 & 6.8 & 0 & 10.3 \\ 0 & 0 & 3.4 & 0 & 0 & 0 & 0 & 6.8 & 0 \end{bmatrix}$$

The distances in the matrix are normalized by the lowest value, which we take to be the social distance 1. Note that the matrix reflects the two-tier nature of the graph. The nodes index the rows of the matrix, from node A representing row 1, to node I representing row 9.

Suppose there are two voice calls competing for resources in the network. Call 1 is being requested by node A for node B. Call 2 is being requested by node F for node G. The social distances for these calls can be found to be:

$$\chi_1 = \min(\chi_{AB}, \chi_{BA}) = 1.6$$

$$\chi_2 = \min(\chi_{FG}, \chi_{GF}) = 6.8$$

Summarizing, the users choose their perceived social distance to all their relationship peers in the network by selecting values from ζ . ζ is kept very simple, and is the same for all the users of the network. Looking at these distances it would not be possible to identify the relative priorities which exist at the global level due to the social network structure. We use the eigenvector centrality of the nodes to rank the users by importance, and compute the modified social distance matrix. The values from this matrix can clearly bring out the scale free nature of social network, and are used in the resource allocation algorithm for prioritization of flows.

2.3 Utility Optimization

Resource allocation problems in networks have been modeled in theory as utility optimization problems [19], with the objective being to optimize the total utility of the network. Utility functions are defined for transport and application layer flows in terms of the network resources allocated to them. They are a mapping from the resource set to the real number scale, with the value of the utility function denoting the relative profit to the flow. Several theoretical utility functions have been defined in literature which achieve a particular objective [20] (e.g., proportionally fair allocation, throughput maximization). At the same time, network protocol implementations (variants of TCP) have been mapped to approximate utility maximization

problems [21].

Utility functions for real-time flows (voice and video) are interesting in that they are subjective, that is they depend on the end user’s opinion about how they perceived the quality of the flow. Thus, these utility functions are derived through user surveys based on several versions of the received (and reconstructed) real-time flow. In defining our social distance aware resource allocation problem (SDA), we first look at a simple theoretical extension of the classical utility maximization problem. We follow this up with a discussion of how the social distance aware component of the modified utility function can be determined for real-time flows (since their utility is by definition subjective).

Let S denote the set of all flows in the network. Then the utility $\{U_s | s \in S\}$ is defined to be,

$$U_s : c_s \rightarrow \mathbb{R} \quad | \quad c_s \subseteq N_R \tag{2.2}$$

where c_s is the subset of network resources N_R currently allocated to flow s . The utility is usually a non-decreasing function of the resources allocated to the flow.

The classical network utility optimization problem (in the case of a wired network) for our example social network (Figure 2.1) is as follows:

$$\begin{aligned} & \underset{\{c_s\}}{\text{maximize}} && \sum_s U_s(c_s) \\ & \text{subject to} && \sum_s c_s \leq N_R \end{aligned} \tag{2.3}$$

Since the utility function of similar flows will be the same, resources are shared equally among the flows without regard to the real-world priority of the flow. In a scenario where the network resources are constrained, the resource allocation according to Problem 2.3 would not be optimal because all the competing flows lose resources equally, when in fact some of them are at a lower priority than the others. Also, the solution to Problem 2.3 under current service differentiation schemes (classifying by traffic type) would produce unfair allocations for scenarios where a high priority flow is using a non-high priority traffic type.

We therefore incorporate the social distance of a flow into the definition of its utility. One way to achieve this is by introducing per-flow social distance dependent utility bounds, which restrict the maximum achievable utility of a flow with increasing resources. The idea behind this is that by suitably modifying the maximum achievable utility (with increasing resources), the original solution algorithm for Problem 2.3 could produce the desired (social distance determined) service differentiation. We note here that for the case of real-time flows, there can

also be a bound on the minimum utility acceptable to the flow so as to create a threshold on the received quality. We denote the maximum utility bound for flow s by U_{max}^s . This bound depends on the social distance of the flow. For lower social distances (high priority) the bound will be high. The mathematical form of U_{max}^s is dependent on U_s as well as χ_s . Thus, the modified utility maximization problem can be defined as:

$$\begin{aligned}
& \underset{\{c_s\}}{\text{maximize}} && \sum_s U_s(c_s) \\
& \text{subject to} && \sum_s c_s \leq N_R \\
& && U_s(c_s) \leq U_{max}^s(\chi_s)
\end{aligned} \tag{2.4}$$

This problem can be decomposed because the maximum utility bound constraint is a per-flow constraint. Dual decomposition of this problem gives us the per-flow optimization problem to be:

$$\begin{aligned}
& \underset{c_s}{\text{maximize}} && (U_s(c_s) - \lambda(U_s(c_s) - U_{max}^s(\chi_s))) - \mu_s(c_s) \\
& \text{subject to} && c_s \in \mathcal{C}
\end{aligned} \tag{2.5}$$

The utility function of the flow is now a function of both the resources allocated to it, as well as the social distance associated with the flow. More generally, in order to incorporate the social distance into resource allocation decisions at the network, we define a modified utility function \hat{U}_s for flow s which depends on c_s as well as χ_s . Let flow s originate at node i and the destination be node j . Then the modified utility function takes the form,

$$\begin{aligned}
\hat{U}_s(c_s, \chi_s) &= U_s(c_s) - \beta f(\chi_s, c_s) \\
&\text{where } \chi_s = \min(\chi_{ij}, \chi_{ji})
\end{aligned} \tag{2.6}$$

This social distance aware utility function can then inform the network of its ability to accept suboptimal resource allocations. Flows with higher social distances do not need the highest resource allocation, and the acceptable allocations can be suboptimal with respect to the original utility function U_s . Through \hat{U}_s we make this fact explicit in the formulation, thereby allowing the original solution algorithm to work without any change.

We now provide a generic example in order to illustrate how social distance affects the optimal resource allocation problem. Let all the flows have the logarithmic utility function

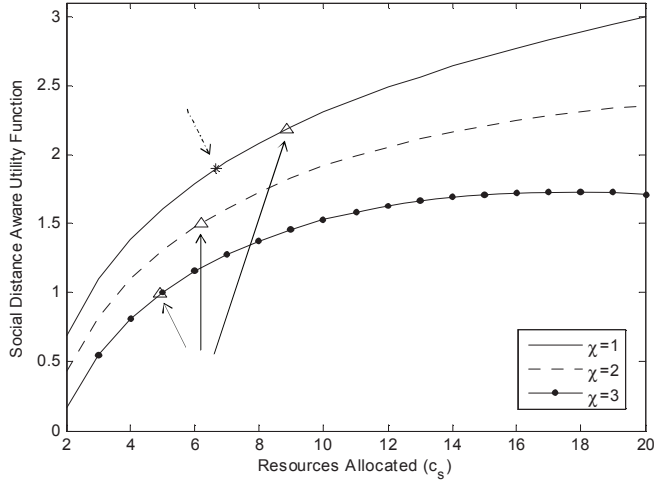


Figure 2.2: Prioritized resource allocation among three flows of the same traffic type.

which results in proportionally fair resource allocation between the flows.

$$U_s(c_s) = \log(c_s) \quad (2.7)$$

We consider $n = 3$ flows in the system. For the case of $N_R = 20$, the optimal solution to Problem 2.3 is $c_s = 6.67$ for $S = \{s_1, s_2, s_3\}$. Now consider the social network graph for the users is such that $\chi(S) = \{1, 2, 3\}$. We define the social distance aware utility function as,

$$\begin{aligned} \widehat{U}_s(c_s, \chi_s) &= \log(c_s) - 0.5 \left((\chi_s - 1) \left(e^{-k} \right) \right) \\ \text{where } k &= \frac{15 - c_s}{2} \end{aligned} \quad (2.8)$$

Such a definition produces scaled utility functions for the same traffic type depending on χ_s . The resulting utility functions and the optimal solutions for the three flows are shown in Figure 2.2. The optimal allocation of resources for the three flows is evaluated to be $\{8.87, 6.21, 4.92\}$. Compare this to the original optimal point for the three flows $\{6.67, 6.67, 6.67\}$, also shown in the figure. The prioritized resource allocation was achieved without changing anything in the solution algorithm.

2.3.1 The function f

The choice of f is necessary to the definition of utility in terms of the social distance. By looking at the form of \widehat{U}_s (2.6) we know that f has to be convex in order for the modified utility function to remain concave. The social distance matrix will remain constant for a much longer timescale than resource allocation decisions for the span of a flow. Thus, χ_s can be assumed to be constant in f for a given flow (a specific source-destination pair). Equation 2.6 can be interpreted as reducing the incremental increase in utility for a unit increase in the allocated resources. Such a definition should produce different versions of the utility function as shown in Figure 2.2. Thus, the exact definition of f in terms of c_s and χ_s depends on U_s , and hence on the traffic type.

For the case of real-time flows, the utility functions are subjective measures dependent on the perception of the end user about received quality. The mean opinion score for voice is one such utility function. Similar functions are defined for video traffic (e.g., peak signal to noise ratio). The utility function is defined in terms of thresholds on received quality such as a mean opinion score of 5 is excellent quality and so on. Such a utility function can also then be used to set bounds on *expected* quality at the receiver depending on the social distance. For example, a call with higher social distance (lower priority) will demand received quality only up to a mean opinion score of 4. Once the acceptable thresholds of the utility function are mapped to corresponding resource demands from the network, we can define f as the function which achieves conversion from social distance to the expected utility bounds.

The assumption that we have such thresholds of the utility function is necessary to accommodate inelastic traffic optimally. The reasoning is that maximizing resource allocation without changing the utility threshold is of no use to the inelastic flow. It can do without the extra resources and still achieve the same perceived quality at the receiver. The extra resources can then be allocated to other inelastic/elastic flows.

The definition of \widehat{U}_s also helps identify the real-world priority of flows irrespective of traffic type. In wireless networks with a MAC layer capable of delivering quality of service (IEEE 802.11e QBSS), voice is prioritized higher than video traffic. This works to the detriment of video flows when the system (QBSS) is working at capacity. Through social distance aware utility functions for both voice and video, we can prioritize flows by their perceived importance rather than traffic type.

The choice of f is guided by the utility function of the flow, and by the mapping from network resources to achievable utility. It will thus be different for each traffic type. It also depends on the social distance matrix of the network. We focus on the specific case of users communicating through voice and video traffic, for the social network shown in Figure 2.1, over a wireless network capable of providing QoS differentiation (QBSS).

2.4 Social Distance in Content

The objective of end users accessing the network can broadly be either to communicate with other users through real-time flows, or to receive content of their interest hosted on the network. For the case of content, the quality of service is determined by the transfer times needed to deliver content to the user. This can be obviously improved for the case of invariant content through replication and distribution at various levels in the network. It is thus important to determine the probability of access for content in order to bring it closer to end users.

Relationships between content data have been used to determine the importance and relevance of individual datum (e.g., web pages) within the content space. This is the predominant feature of search algorithms which rank web pages based on the weight of links which refer to them (incoming links). With the increasing popularity of social media (e.g., Youtube), such relationships between content are now explicit. The set of relationships can actually be used to determine the user's future access patterns. In the case of video content, the data does not change over time and thus is suitable for caching at various levels in the network. The decision to bring content closer to the user is also a part of Traffic Engineering, since it improves the end user's experience of the network without changing his/her access network's resources.

2.4.1 Rank Distribution

The characteristics of user access patterns about related video content, especially for the case of Youtube, have been studied in literature [22][23]. The properties of rank distributions (access counts) for video items have also been studied [24], and compared to the Zipf distribution which has been shown to be applicable to web pages. It is this property of web pages (Zipf distribution) which makes caching attractive.

There are publicly available data traces about Youtube content [25] for videos belonging to particular "categories" in Youtube. We look at the feasibility of caching for video data by studying the rank distributions of the available data traces. The two categories for which traces have been made available are the Science and Technology category, and the Entertainment category.

In the data traces, for each video, information about its popularity, rating, size, duration and several other properties are compiled. We extract the number of views properties for all the videos in these traces to determine the rank distributions for these videos. The results are shown in Figure 2.3. Videos are ranked starting from 1, with the video with the highest number of views ranked as 1. The rank distribution follows power law, which is a characteristic of social networks. It is close to Zipf distribution for static web pages, except that the tail of the curve tapers off exponentially.

The popular videos are all present at the head of the curve, which follows Zipf distribution,

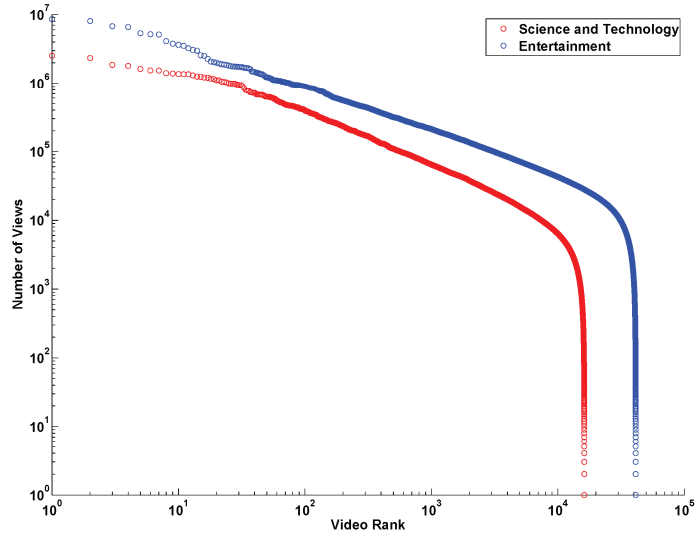


Figure 2.3: Rank distributions for two categories of Youtube videos.

and thus video content is suitable for caching. We determine the social distance between related videos by making it inversely proportional to the popularity of a video. Thus, a highly popular video file will have low cost edges from all its neighbors. The result of this is a social network of videos, structured by social distances which are in turn determined by the popularity of individual videos.

2.4.2 Content Caching and Distribution

For the case of invariant content, distribution and caching of content can improve the network performance by reducing repetitive requests over the network for the same content. It can also improve the perceived end user experience of the network by bringing the content closer to the user, and thus reducing transfer times without changing anything in the user’s access network. We focus on the problem of identifying important content in Chapter 6, and evaluate different centrality measures in terms of their effectiveness in identifying the most queried content over the network.

2.5 Summary

In this chapter, we proposed a way to measure the social distance between users by determining the relative importance that a user attaches to his/her relationships. From such a localized measure of social distance, we showed how a global social-network-wide social distance measure

can be generated by combining the user's preferences with his/her overall importance in the wider social network. The importance of a user is measured in terms of his/her eigenvector centrality.

We then incorporated the social distance measure into the definition of utility function for a communication flow. We initially demonstrated how this could be done through imposing social distance aware utility bounds on flows, and then provided a generic definition of a social distance aware utility function. We also provide an example of social distance aware resource allocation for the case of elastic traffic.

Finally, we looked at the case of social distance between content, and how this can be used to study the relevance/popularity of content in the content space. We showed rank distributions for the case of publicly available data traces for Youtube videos, which demonstrate that video content access also follows the Zipf distribution for the popular videos. This means that the content is a good candidate for caching and distribution, which can be used to improve network performance, as well as improve the end-user's experience of the network.

Chapter 3

Voice Call Capacity Analysis

The original IEEE 802.11 standard for a wireless LAN did not have any capability to provide differentiated services to real-time flows. It is only through the IEEE 802.11e extension to the original standard that such service differentiation can be achieved between flows of differing traffic types. Accommodating real-time flows into a wireless network while meeting their quality of service requirements necessitates the need for estimating capacity of wireless networks for such real-time flows. Unless there is a way to measure when the capacity will be reached, or at least to predict how many flows can be concurrently accommodated while meeting their individual service requirements, some of the flows will fail. Thus, capacity analysis of a wireless network can help provide theoretical thresholds for the number of concurrent real-time flows which can be accommodated.

In this chapter, we look at the problem of developing a closed form equation for wireless channel capacity, for the specific case of voice traffic. It is important to have an estimate of the number of voice calls that can be accommodated in a wireless network such that all the calls receive acceptable service (mean opinion score). We do this for the case of the standard IEEE 802.11b wireless LAN with a maximum data rate of 11Mbps.

The channel capacity will differ based on the achieved data rate of the channel, as well as the source coding which is used for voice calls. We define the voice call capacity of the channel in terms of these variables, and perform simulations for every combination of the variables to determine the achieved call capacity. We then fit the data to our proposed model using linear regression fitting. Using the closed form equation for capacity, we determine the degree of dependence of voice call capacity on the chosen source coding. An explicit knowledge of how the choice of source coding affects voice call capacity can help in choosing the optimal value for this variable.

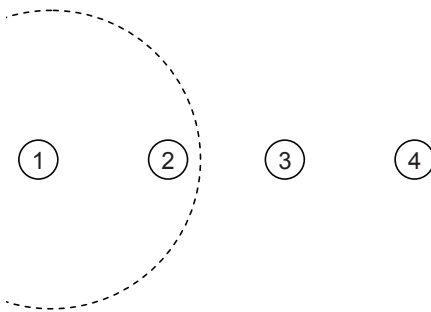


Figure 3.1: The simple chain topology used for our simulations.

3.1 Background

In the literature, voice call capacity has been studied for different wireless network configurations, specifically for the case of a wireless mesh network [26][27]. The focus of these works is on performance optimizations for voice calls such as packet aggregation and header compression for improving the capacity in the case of wireless mesh networks. Another related work [28] deals with the challenges faced by voice calls at different layers of the protocol stack – MAC layer, routing layer and mobility management. There are also studies which look at the QoS challenges and pitfalls for wireless mesh networks based on the 802.16 mesh mode [29].

In a related previous work [30], the authors had looked at the effect of the Wireless LAN on the voice service. Response surface modeling methodology was applied to derive the throughput and voice call capacity for Wireless LAN's. In a later work [31], an extended model which took into account cross layer interactions through layer interface parameters was proposed.

3.2 Simulation Setup

We use the simple chain topology for our simulations (Figure 3.1), with number of hops varying from 1 to 3. The distance between nodes is set to be 250m, and the transmission range is set to be slightly greater than 250m. There are mainly two parameters of interest in determining the voice call capacity. They are the channel data rate, and the voice call packetization interval. The channel data rate is set to be one of $\{1, 2, 5.5, 11\}$. For a n hop topology, this gives us $4n$ possible combinations.

The second parameter of interest is the voice packetization interval. We use the G.711 codec for voice calls [32], which is the pulse code modulation standard for voice traffic. Voice calls encoded using G.711 have a data rate of 64Kbps with each packet containing 10ms of voice data.

This is the basic packetization interval (10ms). It can be changed so as to aggregate more voice samples into the same packet, and such a choice sets up a tradeoff between delay and packet loss. For high packetization intervals, loss of a single packet can result in a significant loss of quality. We vary the packetization interval between 10ms to 100ms for our simulations, which results in a total of 10 possible combinations for voice calls. The relation between packetization interval and the packet stream generated is as follows:

$$\begin{aligned}
 \text{PI} &= 0.01 * k, \quad 1 \leq k < 10 \\
 \text{PacketSize} &= 80 * k \text{ bytes} \\
 \text{IARate} &= 0.01 * k
 \end{aligned} \tag{3.1}$$

where PI represents the chosen packetization interval, and IARate is the inter-arrival rate for voice packets.

The voice traffic is generated using CBR traffic generator in ns-2 with the parameters given by Eq. 3.1, with the duration of the voice calls set to be 3 minutes. We compute the achieved QoS of a given voice call through the mean opinion score. In order to do this, we first compute the R-value of the call [33], and then translate it to the mean opinion score [34].

Starting from a single voice call, we increase the number of calls until at least one call fails. We do this for every combination of data-rate and packetization interval. At the end of the simulations we have the set of system responses for all the different combinations of input parameters. This data forms the input to the SAS GLM procedure to derive the metamodels for call capacity.

3.3 Call Capacity Models

In order to determine voice call capacity models, we first formulate a generic function in terms of all the chosen system variables (which were varied during the simulation). The call capacity of a wireless channel is a non-linear function of our chosen variables, mainly due to the shared broadcast channel coupled with the exponential backoff mechanism. In order to apply linear regression fitting for our proposed call capacity functions (with the simulation data), we transform the variables. We use the natural logarithm of the variables in place of the variable itself, and thus use the logarithm to linearize the non-linear form of channel capacity.

3.3.1 Single Hop Wireless Network

This is the equivalent of the downlink from an AP to the station, and thus determines the capacity of the maximum capacity of the downlink in the absence of any interfering flows. There are only two variables in this scenario, the channel data rate (x_1) and the voice packetization interval (x_2). We define the following sets for easy characterization of these variables:

$$D = \{1, 2, 5.5, 11\} \quad (3.2)$$

$$PI = \{0.01 * k \mid 1 \leq k < 10\} \quad (3.3)$$

$$I_1 = \{1, 2\} \quad (3.4)$$

Now we can define our variables to be $x_1 \in D$, and $x_2 \in PI$. The voice call capacity is defined in terms of x_1 and x_2 as:

$$N = \beta_0 + \sum_{i \in I_1} \beta_i \ln(x_i) + \sum_{i, j \in I_1, j > i} \beta_{(i+j)} \ln(x_i) \ln(x_j) \quad (3.5)$$

We obtain an estimate of the beta's from SAS GLM, and the fitted values are shown in Eq. 3.6. The ANOVA R -square measure for this fit is 0.987911. A value closer to 1 is desired, and thus this is a good fit for the voice call capacity.

$$\begin{aligned} \beta_0 &= 20.203, \beta_1 = 28.490 \\ \beta_2 &= 3.728, \beta_3 = 6.724 \end{aligned} \quad (3.6)$$

3.3.2 Two Hop Wireless Network

This is representative of a single collision domain wireless network (with reference to the AP), where communication between clients is achieved through the AP. In this case, we have two link data rates to consider. We represent these by x_1 and x_2 . The packetization interval is represented by x_3 . The variables can be defined as, $x_1 \in D$, $x_2 \in D$ and $x_3 \in PI$ (Eq. 3.4). We define the following set, $I_2 = \{1, 2, 3\}$. The function for voice call capacity is of the form:

$$\begin{aligned} N &= \beta_0 + \sum_{i \in I_2} \beta_i \ln(x_i) + \sum_{i, j \in I_2, j > i} \beta_{(i+j+1)} \ln(x_i) \ln(x_j) \\ &+ \sum_{i, j, k \in I_2, k > j > i} \beta_{(i+j+k+1)} \ln(x_i) \ln(x_j) \ln(x_k) \end{aligned} \quad (3.7)$$

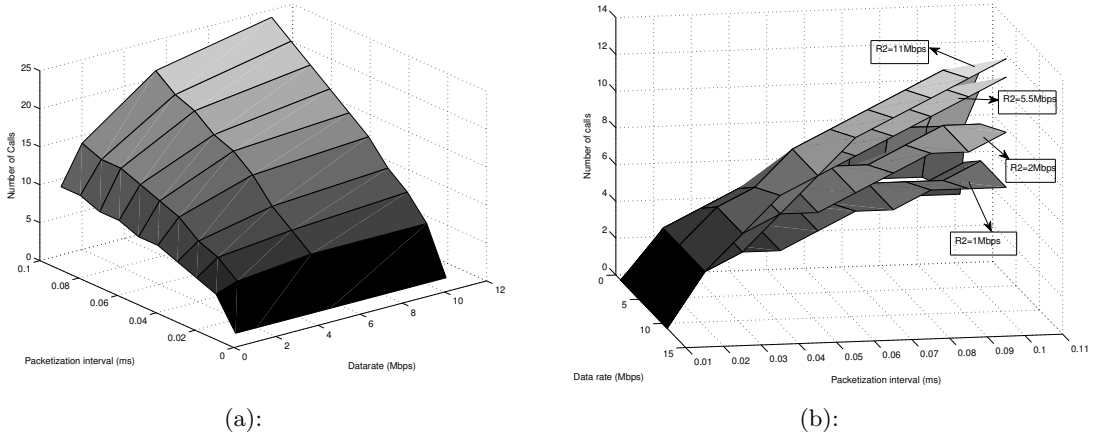


Figure 3.2: System Response for 2 Hops (Figure 3.2a) and 3 Hops (Figure 3.2b) are shown. For both cases, one of the links is set to have the data rate of 11Mbps, while the other data rates are varied.

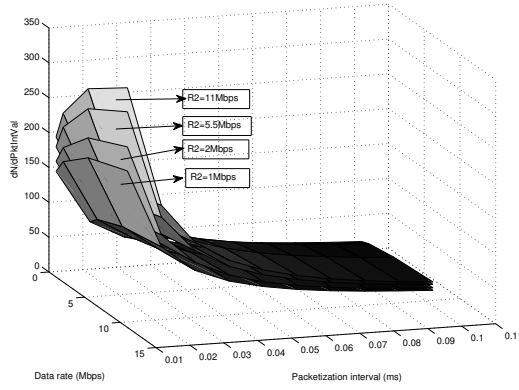
From the SAS GLM procedure, we obtain an estimate of the beta's which are shown in Eq. 3.8. The ANOVA R -square measure for this fit is 0.980833.

$$\begin{aligned}\beta_0 &= 12.339, \beta_1 = 2.983, \beta_2 = 3.114, \beta_3 = 2.531 \\ \beta_4 &= 3.639, \beta_5 = 0.638, \beta_6 = 0.669, \beta_7 = 0.875\end{aligned}\quad (3.8)$$

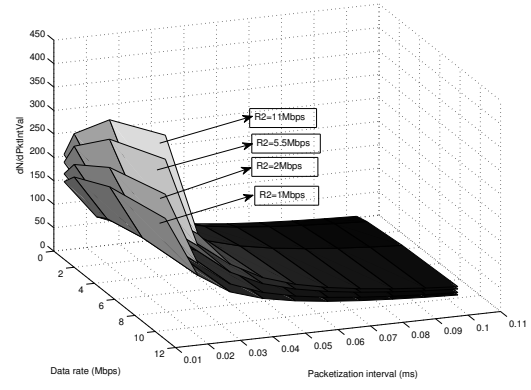
3.3.3 Three Hop Wireless Network

In this scenario, we have 3 link data rates as part of the variables. These are represented by x_1, x_2, x_3 respectively. The packetization interval is represented by x_4 . The variables are defined as follows: $x_1, x_2, x_3 \in D$, $x_4 \in PI$ (Eq. 3.4). The new variable set is defined to be $I_3 = \{1, 2, 3, 4\}$. The function for voice call capacity is of the form:

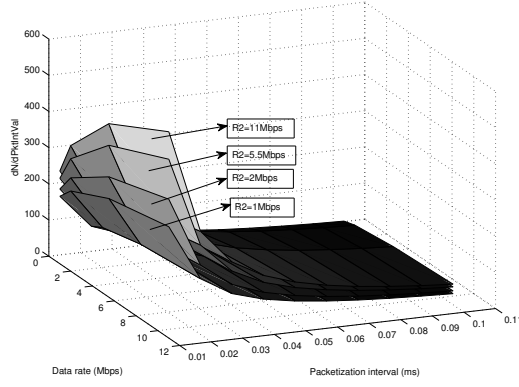
$$\begin{aligned}N &= \beta_0 + \sum_{i \in I_3} \beta_i \ln(x_i) + \sum_{i, j \in I_3, j > i} \beta_{(i, j)} \ln(x_i) \ln(x_j) \\ &+ \sum_{i, j, k \in I_3, k > j > i} \beta_{(i, j, k)} \ln(x_i) \ln(x_j) \ln(x_k) \\ &+ \sum_{i, j, k, l \in I_3, l > k > j > i} \beta_{(i, j, k, l)} \ln(x_i) \ln(x_j) \ln(x_k) \ln(x_l)\end{aligned}\quad (3.9)$$



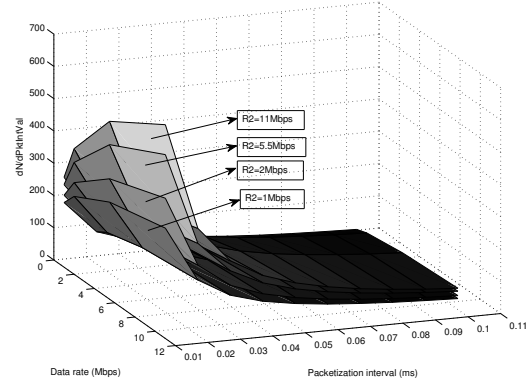
(a): $x_1 = 1\text{Mbps}$



(b): $x_1 = 2\text{Mbps}$



(c): $x_1 = 5.5\text{Mbps}$



(d): $x_1 = 11\text{Mbps}$

Figure 3.3: Partial derivatives of the call capacity (for 3 hops) with respect to the packetization interval.

From the SAS GLM procedure, we obtain an estimate of the beta's which are shown in Eq. 3.10. The ANOVA R -square measure for this fit is 0.972616.

$$\begin{aligned}
 \beta_0 &= 7.335, \beta_1 = 0.898, \beta_2 = 0.905, \beta_3 = 1.393, \beta_4 = 1.516, \beta_{1,2} = 0.256 \\
 \beta_{1,3} &= 0.202, \beta_{1,4} = 0.205, \beta_{2,3} = 0.517, \beta_{2,4} = 0.534, \beta_{3,4} = 0.303, \beta_{1,2,3} = 0.504 \\
 \beta_{1,2,4} &= 0.049, \beta_{1,3,4} = 0.119, \beta_{2,3,4} = 0.121, \beta_{1,2,3,4} = 0.125
 \end{aligned} \tag{3.10}$$

3.4 Sensitivity Analysis of the Call Capacity

Sensitivity analysis of the call capacity with respect to different parameters can be computed using partial derivatives of the system response metamodel. As an example, we focus here on the choice of Packetization interval. We want to observe the effect of choice of packetization interval on the system response and thus compute the partial derivative of N with respect to x_4 for the three hop topology. Using the metamodel built in the previous section, we can calculate the sensitivity w.r.t. x_4 as:

$$\begin{aligned} \frac{\partial N}{\partial x_4} = & \frac{\beta_4}{x_4} + \frac{\beta_{(1,4)} * \ln(x_1)}{x_4} + \frac{\beta_{(2,4)} * \ln(x_2)}{x_4} + \frac{\beta_{(3,4)} * \ln(x_3)}{x_4} \\ & + \frac{\beta_{(1,2,4)} * \ln(x_1) * \ln(x_2)}{x_4} + \frac{\beta_{(1,3,4)} * \ln(x_1) * \ln(x_3)}{x_4} \\ & + \frac{\beta_{(2,3,4)} * \ln(x_2) * \ln(x_3)}{x_4} + \frac{\beta_{(1,2,3,4)} * \ln(x_1) * \ln(x_2) * \ln(x_3)}{x_4} \end{aligned} \quad (3.11)$$

The results are shown in Figure 3.3a, Figure 3.3b, Figure 3.3c and Figure 3.3d.

We can observe from the plots that at lower values of packetization interval (less than 20ms), the parameter affects the system capacity to a larger extent. After the packetization interval increases above 30ms, the effect is not as prominent as it was in the previous case. Thus, packet aggregation at intermediate hops can only be useful when the packets are received at a lower packetization rate (less than 20ms). Aggregation for packetization rates above this *threshold* will not increase the system capacity by a large extent.

3.5 Summary

Call capacity models for multiple hops in a wireless network were determined for the case of voice calls in an IEEE 802.11b network. In deriving these models we used measurements from simulations to generate linear regression fitted functions for call capacity. Depending on the choice of variables in terms of which the call capacity is defined, we can then look at the influence the parameter settings for these variables that vary the achieved call capacity. We looked at one such source coding option for voice calls, namely the packetization interval. The inference is that packetization interval above 20ms does not significantly increase the network call capacity.

Chapter 4

Resource Allocation for Voice Calls

Social distance aware resource allocation can help increase network utility by allocating resources based on social distance for competing flows of the same traffic type. In this chapter we look at the problem of social distance aware resource allocation for voice calls in a wireless network. The wireless network is assumed to be IEEE 802.11e QBSS, capable of differentiating between flows by traffic type.

In the case of wireless networks, channel resource allocation is important due to the broadcast nature of communication. The shared channel forces a contention-driven channel access mechanism with stations performing a backoff to resolve collisions. The MAC layer in a wireless network is thus the prime determinant of the resulting resource allocation to the stations. Since every flow has to contend for the same channel resource, there can be no *guarantees* on the quality of service extended to particular flows. However, service differentiation can be provided by using separate queues for different types of traffic.

4.1 Overview

We first formulate the theoretical utility maximization problem for voice calls in an IEEE 802.11e QBSS. We extend it to social distance aware resource allocation (SDA), and explain the way the highly coupled utility function can be decomposed. The theoretical problem is solved using Matlab for both IEEE 802.11e and SDA. Following this, we present the simulation results for the same scenario which show that the network utility increases due to SDA as compared to standard IEEE 802.11e.

The basic premise of SDA is as shown in Figure 4.1. The network is made aware of the social distance aware utility of the flow, due to which it can allocate differentiated resources to the flows based on their perceived priority. The resource allocation is in fact achieved by correctly setting the IEEE 802.11e QBSS parameters on the nodes, as well as the AP. The value of these

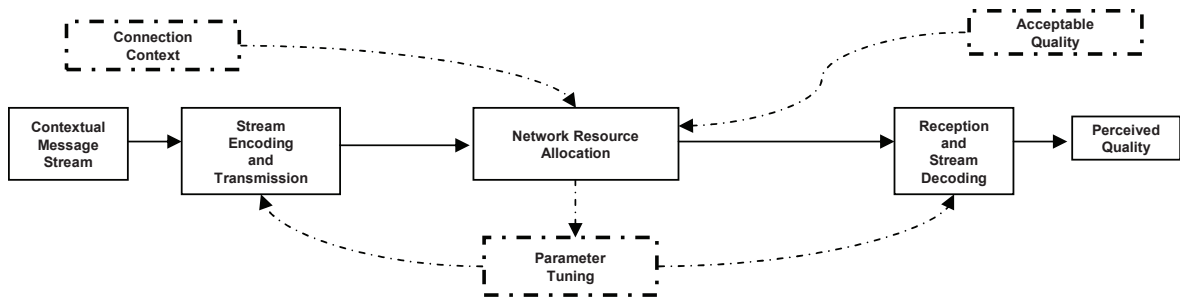


Figure 4.1: The network is made aware of the social distance dependent utility, due to which it can allocate resources differently to flows of the same traffic type. Resource allocation is achieved by setting parameters on the client stations as well as the AP in the case of a wireless network.

parameters are determined statically by IEEE 802.11e, whereas SDA chooses the parameters based on the social distance.

4.2 IEEE 802.11e QBSS and SDA Theoretical Analysis

The IEEE 802.11e extension to the IEEE 802.11 WLAN standard provides for service differentiation using four access categories (AC). The AC's are parameterized, with each AC capable of accepting four parameters. These are the AIFS (Arbitration Inter Frame Spacing) which determines the mandatory wait times for an AC before it can transmit, Contention window (CWMin, CWMax) values which determine the length of a backoff duration, and TXOP which determines the number of consecutive packets that can be transmitted after a successful contention period. We only focus on the first three parameters for our work.

While AC based channel access can distinguish between voice and best effort traffic, it cannot make a distinction between two voice flows with different relative priorities. This is however possible if we use the social distance associated with the voice flows to distinguish them. Video flows are treated at a lower priority than voice due to the AC separation. Thus, when the wireless network is working at capacity, video flows are starved when compared to voice calls. Allocating AC's by the social distance can overcome both these shortcomings. In the following two sections, we compare IEEE 802.11e with SDA in terms of their resource allocation algorithms.

In this section, we provide a theoretical formulation of SDA for voice calls, solve the resulting optimization problem using Matlab and compare the achieved network utility with standard IEEE 802.11e. In the next section, we provide simulation results comparing the performance

of IEEE 802.11e and SDA when voice traffic competes for resources with best-effort traffic.

4.2.1 Utility Maximization

Consider the case where the users from Figure 2.1 communicate through an IEEE 802.11e QoS WLAN (QBSS), with the IEEE 802.11g PHY supporting data rates up to a maximum of 54Mbps. The users communicate using voice calls through the wireless network. There is HTTP traffic present in the network which uses the best-effort AC. We will formulate a utility optimization problem for voice calls, with a social distance aware utility function defined for voice traffic, to determine AC parameters for each flow depending on the social distance.

The utility function of a voice call is defined in terms of the latency and packet loss ratio of the call [33][34]. The latency of a call is given by the average access delay of the voice AC, which depends not only on parameters of the AC itself, but also on those of all other AC's. It is also dependent on the number of flows in each AC competing for the channel. Thus, the voice utility is highly coupled with the AC parameters of all the flows in the network. For simplicity of presentation, we only look at determining the AIFS parameter in this section. Let the utility of a voice call i in the system be represented by $U_i(AIFS_i, \{AIFS\}_{j \neq i})$.

In order to accommodate the social distance into resource allocation decisions, we tie the social distance to expected minimum ($U_{min}^{vo}(\chi_i)$) and maximum ($U_{max}^{vo}(\chi_i)$) utility bounds on the voice flow.

$$\begin{aligned} U_{min}^{vo}(\chi_i) &> U_{min}^{vo}(\chi_j) && \forall \chi_i < \chi_j \\ U_{max}^{vo}(\chi_i) &> U_{max}^{vo}(\chi_j) && \forall \chi_i < \chi_j \end{aligned}$$

With the voice utility expressed in terms of the mean opinion score (value from 1 to 5, with 5 representing the best experience), U_{max}^{vo} defines the best and U_{min}^{vo} the least acceptable values for the mean opinion score. These will differ between calls depending on the social distance because not all calls are equally important.

The utility maximization problem for voice calls in the network can be formulated as:

$$\begin{aligned} &\underset{\{AIFS_i\}}{\text{maximize}} && \sum_i U_i(AIFS_i, \{AIFS\}_{j \neq i}) \\ &\text{subject to} && AIFS_i \in VO_AC \\ & && U_i \leq U_{max}^{vo}(\chi_i) \\ & && U_i \geq U_{min}^{vo}(\chi_i) \end{aligned} \tag{4.1}$$

The set VO_AC contains AIFS values $\{2, 3, 4, 5, 6, 7\}$ for our SDA problem, instead of just $\{2\}$ as specified by the IEEE 802.11e standard.

Problem 4.1 would easily decompose into sub-problems at each user if the utility function was not so highly coupled. A dual-decomposition could have been used to solve the problem efficiently.

In order to eliminate the high coupling in the utility function, the preferred solution is to introduce local copies of the global variables [35][36]. New equality constraints are introduced to correspond to the original problem, so that the local and global variables take the same value in any feasible solution. The problem thus becomes:

$$\begin{aligned}
& \underset{\{AIFS_i\}}{\text{maximize}} && \sum_i U_i(AIFS_i, \{AIFS\}_{ij|j \neq i}) \\
& \text{subject to} && AIFS_i \in VO_AC \\
& && U_i \leq U_{max}^{vo}(\chi_i) \\
& && U_i \geq U_{min}^{vo}(\chi_i) \\
& && AIFS_{ij} = AIFS_j \quad \forall j \neq i
\end{aligned} \tag{4.2}$$

The problem can now be decoupled, and can be simplified using a dual decomposition approach.

$$\begin{aligned}
& \underset{\{AIFS_i\}}{\text{maximize}} && \sum_i U_i(AIFS_i, \{AIFS\}_{ij|j \neq i}) \\
& && + \sum_{i,j \neq i} \gamma_{ij} (AIFS_j - AIFS_{ij}) \\
& \text{subject to} && AIFS_i \in VO_AC \\
& && U_i \leq U_{max}^{vo}(\chi_i) \\
& && U_i \geq U_{min}^{vo}(\chi_i)
\end{aligned} \tag{4.3}$$

The γ_{ij} 's are called consistency prices. The original problem decouples into per-flow sub-problems involving only local variables of the flow.

$$\begin{aligned}
& \underset{\{AIFS_i\}}{\text{maximize}} && U_i(AIFS_i, \{AIFS\}_{ij|j \neq i}) \\
& && + \left(\sum_{j \neq i} \gamma_{ji} \right) (AIFS_i) - \sum_{j \neq i} \gamma_{ij} (AIFS_{ij}) \\
& \text{subject to} && AIFS_i \in VO_AC \\
& && U_i \leq U_{max}^{vo}(\chi_i) \\
& && U_i \geq U_{min}^{vo}(\chi_i)
\end{aligned} \tag{4.4}$$

The master dual problem, $g(\{\gamma_{ij}\})$, is to minimize the value of the Lagrangian in Problem 4.3 for a given set of γ_{ij} 's.

$$\underset{\{\gamma_{ij}\}}{\text{minimize}} \quad g(\{\gamma_{ij}\}) \tag{4.5}$$

The master problem is solved using the following update of the consistency prices:

$$\gamma_{ij}(t+1) = \gamma_{ij}(t) - \alpha(AIFS_j - AIFS_{ij}) \tag{4.6}$$

Due to the local copies of global variables, the per-flow optimization problem (Problem 4.4) is fully determined and can be solved if utility of the flow can be computed. For this, the latency experienced by the call, given by the average access delay of the AC, needs to be determined.

For a given set of AIFS values (own AIFS, and local copies of global variables) the average access delay of the AC is still a non-linear function of the parameters of all AC's in the system. This is because the access delay depends on the collision probability, as well as the average backoff duration of the AC. Both of these parameters depend on the state of the system, that is the number of flows competing for resources at each AC, and the AC parameters themselves. There are several theoretical models which have been proposed to estimate the average access delay of an AC [37, 38, 39, 40, 41]. We implement one such proposed approximation [41], where the average access delay for an AC k is given by,

$$d_k \approx \left(\frac{2c_k \Gamma}{p_k q_k} \right) \tag{4.7}$$

where c_k is the collision probability of AC k , p_k is the transmission probability of AC k ,

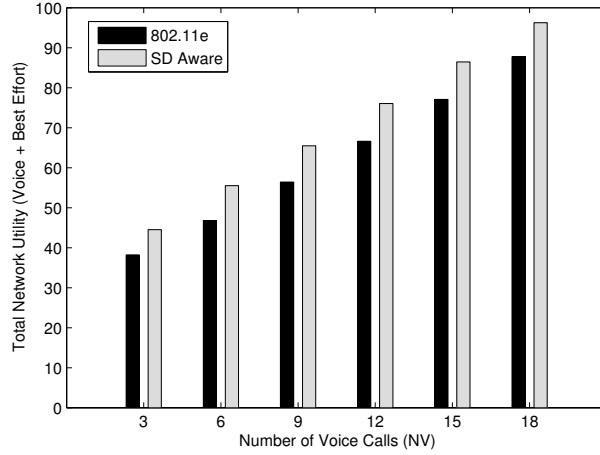


Figure 4.2: Comparison of the total network utility achieved for IEEE 802.11e vs Social Distance Aware Resource Allocation (SDA). SDA outperforms IEEE 802.11e for every case.

and:

$$\begin{aligned}
 \Gamma &= \{AIFS_1 + t_{data} + SIFS + t_{ack}\} \\
 q_1 &= 1 \\
 q_k &= \left(\dots \left((1 - p_1)^{n_1 h_1} (1 - p_2)^{n_2} \right)^{h_2 - h_1} \dots \right. \\
 &\quad \left. (1 - p_{k-1})^{n_{k-1}} \right)^{h_k - h_{k-1}} \\
 h_k &= \left(\frac{AIFS_k - AIFS_1}{t_{slot}} \right)
 \end{aligned} \tag{4.8}$$

The collision and transmission probabilities are expressed in terms of each other in a non-linear fashion [41]. These are solved numerically using Matlab for our theoretical results. However, an access point (AP) can accurately track the collision probability of each AC since it is a local measure. The equations for p_i and q_i can then be solved very quickly and the delay estimated. It should be noted however that the theoretical access delay models are derived for the case of saturation traffic conditions.

4.2.2 Comparing IEEE 802.11e and SDA

We can solve the optimization problem in (4.3) for SDA if we know the call distribution pattern for various social distances. To do this we first classify user flows into social distance classes as follows:

$$\begin{aligned}
\chi^1 &= \{\chi_s \mid 1 \leq \chi_s < 4\} \\
\chi^2 &= \{\chi_s \mid 4 \leq \chi_s < 7\} \\
\chi^3 &= \{\chi_s \mid 7 \leq \chi_s < 11\}
\end{aligned} \tag{4.9}$$

The number of voice calls are assumed to be distributed among the social distance classes as:

$$\chi^1 : \chi^2 : \chi^3 = 3 : 2 : 1 \tag{4.10}$$

In SDA, the highest priority flows will be mapped to the original VO_AC (of the IEEE 802.11e standard). The other flows will be mapped to lower AC's depending on their utility bounds. For the case of IEEE 802.11e, VO_AC is set to $\{2\}$, and the utility bounds are all set to the maximum and minimum (feasible) mean opinion score values. For SDA, VO_AC is set to $\{2, 3, 4, 5, 6, 7\}$, and the utility bounds for the social distance classes are set to be:

$$\begin{aligned}
\chi^1 : \quad U_{min}^{vo}(\chi_s) &= 3.8, \quad U_{max}^{vo}(\chi_s) = 4.5 \\
\chi^2 : \quad U_{min}^{vo}(\chi_s) &= 3.5, \quad U_{max}^{vo}(\chi_s) = 4.1 \\
\chi^3 : \quad U_{min}^{vo}(\chi_s) &= 3.0, \quad U_{max}^{vo}(\chi_s) = 4.1
\end{aligned} \tag{4.11}$$

We consider a total of 7 best-effort flows competing for resources along with the voice calls. For a best effort flow, the utility is measured as $\log(x_s)$, where x_s is the datarate achieved by the flow. The total utility of the network is measured as a sum of utilities of all flows in the network.

We solve the optimization problem in (4.3) for the above parameter settings using Matlab. The results of resource allocation through IEEE 802.11e and SDA are shown in Figure 4.2, which compares the total network utility achieved. We see that allocating resources according to social distance outperforms standard IEEE 802.11e in every instance. The reason is that fewer resources are allocated to voice calls at a higher social distance, thereby allowing the best effort traffic to achieve better datarates, thus increasing the total network utility.

The detailed results are presented in Table 4.1. As we can see, under IEEE 802.11e all voice calls achieve the highest possible mean opinion score, with the best effort traffic getting fewer resources as the voice calls increase. The theoretical delay model predicts a slight increase in the average access delay for VO_AC with increasing number of voice calls, which is reflected

Table 4.1: Network performance comparison for IEEE 802.11e vs SDA when voice calls are competing with best-effort traffic.

Number of Voice Calls	IEEE 802.11e		Social Distance Aware			
	Avg MOS	Best Effort	$1 \leq \chi < 4$	$4 \leq \chi < 7$	$7 \leq \chi < 11$	Best Effort
3	4.1141	3.6980	4.1157	4.0350	NA	4.6303
6	4.1103	3.1592	4.1140	3.9612	3.9612	4.4699
9	4.1078	2.7785	4.1114	3.7208	3.7208	4.2940
12	4.1060	2.4769	4.1060	4.0400	4.0400	3.8820
15	4.1049	2.2176	4.1049	3.9830	3.9830	3.6766
18	4.1042	1.9820	4.1042	3.8881	3.8881	3.4742

in the marginal decrease in achieved mean opinion score values. The best-effort flows see a marked increase in their average access delay (and correspondingly reduced throughput) as the number of voice calls increase (Equation 4.7). Thus IEEE 802.11e allocates resources to voice calls at the expense of best-effort flows. It also tries to allocate equal resources to all the voice calls in the system.

In SDA, we can see that voice calls are differentiated based on their social distance, thereby allowing best effort traffic a comparably higher share of the resources (compared to IEEE 802.11e). The utility bounds allow SDA to determine an optimal AIFS value for each voice call depending on χ_s , thereby allocating differentiated resources to the calls. Since the calls get distributed over different AC's, the increase in average access delay for an AC is also limited. This can be seen in the utilities achieved by best-effort flows under SDA. According to our chosen call distribution, 50% of all voice calls are high priority, while the other 50% can be allocated to lower AC's. We see that calls in χ^2 and χ^3 achieve lower mean opinion scores, but they are still within the bounds we set for the calls.

SDA consistently achieves higher network utility than IEEE 802.11e (Figure 4.2) according to our theoretical results. The difference in average utility for a best-effort flow gets multiplied by a factor of 7, since that is the number of best-effort flows in the system. This increase in utility is greater than the decrease in utility of voice calls, due to allocation at lower AC's. Considering that the higher social distance calls expected the lower utility (expressed through utility bounds), we have increased the network utility by allocating resources according to the social distance rather than simply focusing on the traffic type.

Table 4.2: VoIP traffic R -value, MOS, and distance modified MOS

R-value	MOS	MOS*(R)		
		$\chi = 1$	$\chi = 2$	$\chi = 3$
90	4.339	4.339	4.589	4.839
80	4	4	4.274	4.524
70	3.597	3.597	3.847	4.097
60	3.1	3.1	3.35	3.6
50	2.575	2.575	2.825	3.075
40	2.06	2.06	2.314	2.564
30	1.609	1.609	1.859	2.109
20	1.252	1.252	1.502	1.752
10	1.035	1.035	1.285	1.535
0	1.0	1.0	1.25	1.5

4.3 IEEE 802.11e and SDA Simulation

In this section, we look at simulation results comparing IEEE 802.11e and SDA. The results are based on simulations done using ns-2.

4.3.1 Simulation Setup

The traffic stream generated by VoIP sources is CBR, with packet size determined by the parameter of voice packetization interval. It can vary from $10ms$ to $100ms$. The quality requirements for a voice call are those of less delay ($< 100ms$) and less packet loss ($< 3\%$). The utility of the received stream is computed using the mean opinion score (MOS) [42].

A MOS value greater than 4.5 represents excellent quality, 4 is fair, 3 is annoying, 2 is very annoying, and 1 represents that no communication is possible. We let the social context between users to be represented by an element from the set $\{1, 2, 3\}$, with $\chi = 1$ representing the closest distance or in other words the highest priority call. This means that a call with $\chi = 1$ is very important to the users, and optimum resources need to be allocated to this call. We modify the MOS to include the social distance χ as follows:

$$MOS^*(R) = MOS(R) - \beta(\chi - 1) \quad (4.12)$$

where β represents the effect of distance on acceptable quality. It can be seen that equation Eq. 4.12 is a special case of equation Eq. 2.6. Table 4.2 gives the modified MOS values for the case of $\beta = -0.25$.

Algorithm 1: Resource Allocation Algorithm at AP

```

1: procedure ALGORITHM
2:   INPUT: New call request for  $\chi_j$ .
3:   INIT: Get current  $N_i$  calls for  $\chi_i$ ,  $i=\{1,2,3\}$ .
4:   Solve Allocation, let  $AIFS_j$  be the optimal value.
5:   Compute new delay and loss values:
6:    $x = N_1, y = N_2, z = N_3, ax = AIFS_1$  etc
7:   For  $D, p_x = D[1][ax]/D[x][ax]$ 
8:   For  $L, p_x = L[x][ax]$ 
9:    $p_x = p_{x+y+z}^{ax} + \left(\frac{x}{x+y}\right) p_{x+y}^{ay} + \left(\frac{x}{x+y+z}\right) p_{x+y+z}^{az}$ 
10:   $p_y = p_{x+y+z}^{ay} - \left(\frac{x}{x+y}\right) p_x p_{x+y}^{ay} + \left(\frac{y}{x+y+z}\right) p_{x+y+z}^{az}$ 
11:   $p_z = p_{x+y+z}^{az} - \frac{x}{x+y+z} p_x p_{x+y+z}^{az} - \frac{y}{x+y+z} p_y p_{x+y+z}^{az}$ 
12:   $d_x = \frac{D[1][ax]}{p_x}, l_x = p_x$ 
13:  Compute new utilities  $U_i$  for all calls
14:  IF ( $U_i < 4.0$ ) return "Do not accept call"
15:  return Assign  $AIFS_j$  to call.
16: end procedure

```

We look at the control of MAC layer parameter AIFS for selective resource allocation. We consider the case of an 802.11b WLAN operating at 11Mbps. We first determine the utility of VoIP traffic in terms of the AIFS value assigned to the voice AC (VO AC) through extensive ns-2 simulations.

4.3.2 Utility in terms of AIFS

The resource allocation decision will be based on the following formula:

$$\text{maximize} \quad U(d, l, \chi) - \text{cost}(d, l, \chi)$$

The delay and loss in the equation above are not directly controllable parameters in a WLAN. We thus focus on the MAC layer parameter AIFS, and use it as the control parameter in our resource allocation problem. AIFS, CWMin and CWMax together determine the average access delay for a packet, with AIFS having the most pronounced effect. The resource allocation decision is now based on:

$$\begin{aligned} &\text{maximize} \quad U(AIFS, \chi) - \text{cost}(AIFS, \chi) \\ &\text{subject to} \quad 1 \leq AIFS \leq 7 \\ &\quad \quad \quad 1 \leq \chi \leq 3 \end{aligned}$$

We refer to this henceforth as the *Allocation* problem. The function $U(AIFS, \chi)$ can be evaluated if the values of average end-to-end delay and loss can be estimated from AIFS. We thus compute the delay and loss matrices using extensive simulations in ns-2.

VoIP traffic will have the highest priority in an 802.11e WLAN, but there will always be some background traffic which competes with the voice flows for resources. Thus, for all our simulations we consider background HTTP traffic to be present. HTTP traffic generators are provided as part of the ns-2 simulator. The synthetic HTTP traffic that we use is generated using PackMime [43] in ns-2. The important parameter for this traffic generator is the number of new connection requests per second which arrive at the HTTP servers. We use a connection rate of 5 connections/sec for our simulations. We assign HTTP traffic to the BK AC, with a constant AIFS value of 15. The $\{CWMin, CWMax\}$ values for BK are set to be $\{31, 1023\}$ respectively. For the VO AC, we assign $\{CWMin, CWMax\}$ to be $\{7, 15\}$ respectively.

The delay (D) and loss (L) matrices are as shown below. The columns are indexed by the AIFS value ranging from 1 to 7, and the rows are indexed by the number of calls in the system ranging from 1 to 10. Note that the values are only for successfully placed calls at a particular AIFS for a given number of calls.

$$D = \begin{bmatrix} 0.0015 & 0.0018 & 0.0021 & 0.0024 & 0.0028 & 0.0036 & 0.0038 \\ 0.0021 & 0.0025 & 0.0029 & 0.0033 & 0.0039 & 0.0045 & 0.0051 \\ 0.0046 & 0.0035 & 0.0178 & 0.0177 & 0.0084 & 0.0138 & 0.0185 \\ 0.0225 & 0.0099 & 0.0211 & 0.0060 & 0.0079 & 0.0134 & 0.0176 \\ 0.0435 & 0.0148 & 0.0116 & 0.0095 & 0.0093 & 0.0144 & 0.0322 \\ 0.0202 & 0.0080 & 0.0091 & 0.0087 & 0.0145 & 0.0413 & 0.0641 \\ 0.0104 & 0.0087 & 0.0133 & 0.0161 & 0.0398 & 0.0820 & 0.1123 \\ 0.0162 & 0.0130 & 0.0147 & 0.0413 & 0.0950 & 0.1316 & 0.1645 \\ 0.0163 & 0.0140 & 0.0386 & 0.0888 & 0.1199 & 0.1594 & 0.1626 \\ 0.0115 & 0.0347 & 0.0915 & 0.1448 & 0.1368 & 0.1810 & 0.2464 \end{bmatrix}$$

$$L = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0.0000 & 0.0000 \\ 0.0001 & 0.0001 & 0.0000 & 0 & 0.0012 & 0.0000 & 0.0000 \\ 0.0008 & 0.0001 & 0.0092 & 0.0053 & 0.0022 & 0.0053 & 0.0033 \\ 0.0113 & 0.0054 & 0.0084 & 0.0006 & 0.0025 & 0.0032 & 0.0019 \\ 0.0251 & 0.0118 & 0.0044 & 0.0027 & 0.0046 & 0.0033 & 0.0050 \\ 0.0075 & 0.0024 & 0.0025 & 0.0017 & 0.0031 & 0.0166 & 0.0240 \\ 0.0049 & 0.0025 & 0.0043 & 0.0040 & 0.0141 & 0.0339 & 0.0318 \\ 0.0045 & 0.0043 & 0.0030 & 0.0217 & 0.0223 & 0.0430 & 0.0512 \\ 0.0044 & 0.0041 & 0.0232 & 0.0283 & 0.0243 & 0.0557 & 0.0632 \\ 0.0024 & 0.0177 & 0.0427 & 0.0428 & 0.0346 & 0.0988 & 0.0593 \end{bmatrix}$$

The utility of a VoIP call can now be computed using the definition for MOS (equation 4.12). The delay and loss values may need to be derived from the matrices D and L when the system has a mix of calls belonging to different χ . The formulae for doing this are presented as part of Algorithm 1.

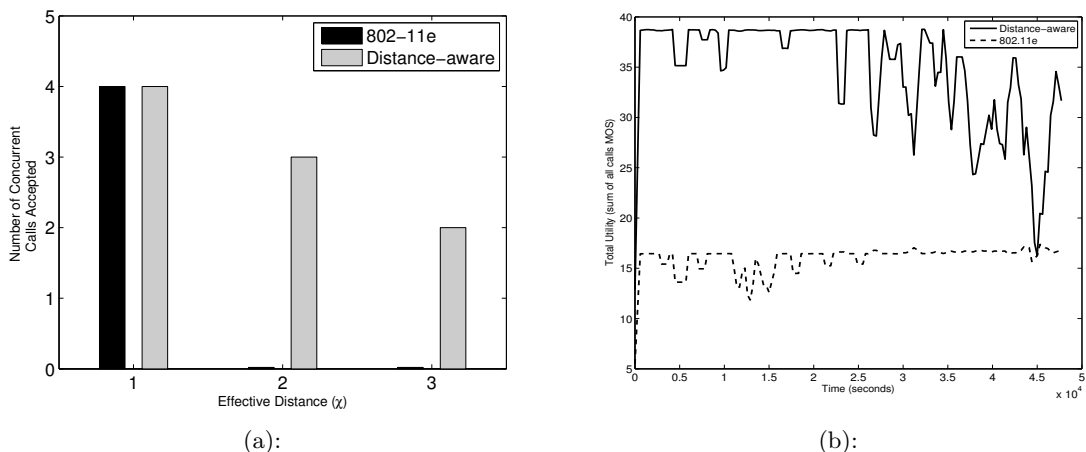


Figure 4.3: Maximum Call Capacity, 802.11e vs Distance-aware. 802.11e VoIP call capacity saturates at 4 calls, hence it can accommodate no calls at distances of 2 and 3 respectively. Total network utility for VoIP calls, plain 802.11e vs distance-aware. As it can be seen, the network can almost double its utility to the users.

We define the cost function to be the parabolic function:

$$cost(AIFS, \chi) = \frac{(AIFS - 2\chi)^2}{2} + 1$$

This choice of the cost function helps distribute the calls from distinct χ values to distinct AIFS values.

4.3.3 Social Graph

To obtain the conditions of a social-connectivity graph, we used a combination of social mobility trace generators and Second Life traces. The social community model was created with the CMM (community mobility model) [44]. Since we are considering the *single collision* domain case, mobility does not have much of an impact on the achievable utility. The social distance was calculated using Second Life traces, which we assume is imposed on the WLAN as an overlay. Furthermore, we apply the inverse-square law for voice intensity to determine thresholds at which voice communication should be possible. For all the results tabulated below, we considered a scenario with 15 nodes situated in a grid of $350m \times 350m$.

4.3.4 Comparing IEEE 802.11e and SDA

802.11e assigns the constant value of 2 to the AIFS of VO AC. SDA assigns calls with different social distance to different AIFS. Figure 4.3a shows that 802.11e call capacity saturates at a maximum of 4 voice calls in the case of 802.11b 11Mbps links. SDA can accommodate 4 voice calls at $\chi = 1$ (as in the case of 802.11e), 3 voice calls at $\chi = 2$ and 2 voice calls at $\chi = 3$ concurrently. This effectively means that we have increased the capacity to 9 voice calls by being aware of the social distance associated with the voice calls.

This also means that the achievable network utility for VoIP calls (sum of MOS of all calls) is always greater in the distance-aware algorithm than that of 802.11e. This can be seen in Figure 4.3b which plots the total utility for the Second Life trace that we are using. The variability is due to the instantaneous mix of calls with different χ competing for the resources.

4.4 Summary

In this chapter, we looked at social distance aware resource allocation for the case of voice calls competing for resources in an IEEE 802.11e QBSS. We defined the social distance aware utility function for voice calls by imposing utility bounds on flows based on the social distance. Since the utility of voice flows is measured in terms of mean opinion score, imposing such utility bounds is eventually subjective and can be interpreted to mean that for a high social distance call (low priority), good call quality can be taken to be the best and so forth.

We formulated the theoretical utility maximization problem for voice calls in a wireless network. By using the social distance aware utility function, this same problem becomes the social distance aware resource allocation problem. The original problem is the model for IEEE 802.11e QBSS. We solve both these optimization problems (after necessary transformation of the problem to decouple it), and compare the achieved network utility for IEEE 802.11e and SDA. Our results show that SDA outperforms standard IEEE 802.11e for every case considered. We also present simulation results for IEEE 802.11e and SDA for the case of voice calls competing for resources with best effort traffic. Our results show that SDA outperforms IEEE 802.11e in terms of achieved network utility by accommodating more voice calls for the same resources.

Chapter 5

Resource Allocation for Voice and Video Flows

According to the IEEE 802.11e QoS standard, voice traffic is always considered high priority compared to video flows. This means that in scenarios where traffic is approaching system capacity, voice flows can effectively starve video flows out of resources. Differentiating solely based on the traffic type (prioritizing real-time over non real-time, voice over video), the network provides preferential access to flows which actually may be low priority when viewed along the social distance dimension. Allocating resources using SDA enables differentiation of flows based on their social distance (for the case of real-time flows) rather than solely based on traffic type.

In this chapter, we present simulation results comparing performance of IEEE 802.11e and SDA when a mix of voice and video calls are competing for network resources in the presence of best-effort HTTP traffic. The utility function for video flows does not have a closed form equation, and thus we do not present theoretical comparisons for the case of voice and video traffic competing for resources. Our simulation results show that SDA, while allocating resources according to social distance, does not unfairly allocate resources solely based on the traffic type when network resources are scarce. Under SDA, the aggregate network utility as well as the saturation capacity are higher than that of standard IEEE 802.11e.

5.1 Using Social Distance for Voice and Video Traffic

The social distance aware utility function for voice was defined in the previous section using maximum and minimum utility bounds on the mean opinion score. This was possible due to the subjective interpretation of received quality by the end user in terms of the mean opinion score. For the case of video traffic, the utility is measured using peak signal to noise ratio (PSNR). The utility of a video flow is also dependent on user perception of received quality, and as such

is a subjective metric. We convert the achieved PSNR for video flows into corresponding mean opinion score values (as in the case of voice calls). On this mean opinion score, we can then define social distance aware utility functions for video using maximum and minimum utility bounds.

We begin with a look at the simulation setup, followed by a detailed discussion of the results for IEEE 802.11e and SDA. In order to further clarify the difference between the two, we present resource allocation patterns of both IEEE 802.11e and SDA for a specific case of voice and video flows competing together for resources.

5.1.1 Simulation Setup

The results in this section are based on simulations done using the IEEE 802.11e model in ns-2. The network topology used for the simulations is shown in Figure 5.1a. The distance from any station to the AP is 100m, and the transmission range of each station is 250m. All the nodes (and the AP) are using the IEEE 802.11g PHY with a datarate of 54Mbps.

In order to simulate SDA using ns-2, we modify the IEEE 802.11e model to accommodate 8 AC's as shown in Figure 5.1b. The social distance from our example network (Figure 2.1) is mapped to the AC's in Figure 5.1b for voice (video) traffic as:

$$\begin{aligned}
 \chi^1 &\rightarrow \text{VO (VI)} \\
 \chi^2 &\rightarrow \text{VO(2) (VI(2))} \\
 \chi^3 &\rightarrow \text{VO(3) (VI(3))}
 \end{aligned} \tag{5.1}$$

Voice calls are assumed to be using the G.711 [32] codec with a 20ms packetization interval. This is simulated using a CBR traffic generator with a data rate of 64kbps and packet size of 160 bytes. The utility of a voice call is measured using the mean opinion score. This is computed by first determining the R-value of the voice call [33], and then converting it to the mean opinion score [34].

For video flows, we use the Foreman video trace [45] for all simulated video flows. We use the MPEG codec to transform the original YUV for transmission. The utility of a video flow is computed using the PSNR (peak signal to noise ratio) metric. PSNR is computed on the received, and decoded video flow as compared to the original YUV. We map the PSNR to a mean opinion score, so that we have a consistent measure to compute the total network utility. It also facilitates the use of utility bounds for video, as in the case of voice (4.1). We use the following mapping between PSNR and mean opinion score:

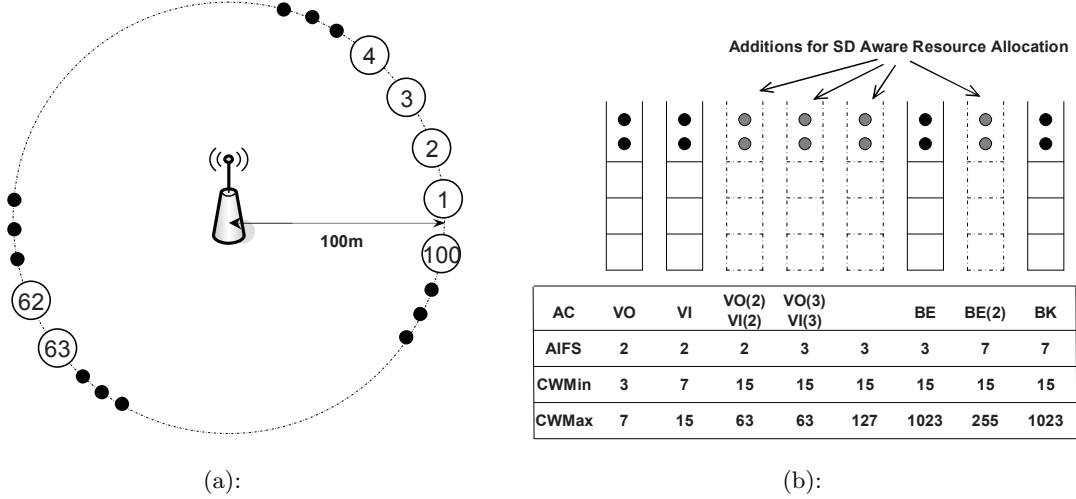


Figure 5.1: The network topology used for simulation is shown in 5.1a. Nodes are placed around a circle of radius 100m with the AP at the centre. Transmission range of all the nodes is set to 250m. 5.1b shows the modification of IEEE 802.11e AC’s to include social distance aware traffic AC’s. The corresponding AC parameters are also listed.

$$\begin{aligned}
 37 &\leq \text{PSNR} < 40 \Rightarrow \text{MOS} = 5 \\
 31 &\leq \text{PSNR} < 37 \Rightarrow \text{MOS} = 4 \\
 25 &\leq \text{PSNR} < 31 \Rightarrow \text{MOS} = 3 \\
 20 &\leq \text{PSNR} < 25 \Rightarrow \text{MOS} = 2 \\
 \text{PSNR} &< 20 \Rightarrow \text{MOS} = 1
 \end{aligned}
 \tag{5.2}$$

We use PackMime [43] to simulate the best effort HTTP flows. Three PackMime streams are initialized in the network at the Best Effort AC. The average connection rate for each stream is set to 1.5, which is the rate at which new HTTP connections arrive every second. In order to measure the utility of the HTTP flows, we use the logarithmic function, $\log(x_s)$, where x_s is the datarate achieved by the best effort flow s .

The distribution of the social distance for voice (video) flows is given by Equation 4.10. The total network utility is computed as the sum of the utilities of voice, video and best effort traffic achieved during a simulation run.

Table 5.1: IEEE 802.11e vs SDA performance comparison for a traffic mix consisting of both voice and video flows.

	NV,NVI	Voice		Video		Best Effort	Total Utility
		\sum MOS	# Calls	\sum MOS	# Flows	$\sum \log(x_s)$	
IEEE 802.11e	1,2	4.11642	1	7.84155	2	23.8336	35.7916
	2,4	8.21285	2	15.6706	4	22.1613	46.0448
	3,6	12.2881	3	23.4463	6	17.639	53.3734
	4,8	16.3536	4	30.8995	8	11.7685	59.0216
	5,10	20.3642	5	7.84155	2	10.6247	38.83045
	6,12	24.2192	6	1.8881	0	10.8212	36.9285
SDA	1,2	4.11643	1	7.84155	2	23.8639	35.8219
	2,4	8.21018	2	15.6831	4	22.1682	46.0615
	3,6	12.2793	3	23.522	6	17.3755	53.1768
	4,8	12.2084	3	30.5683	8	12.6276	55.4043
	5,10	12.2949	3	31.2079	8	10.9737	54.4765
	6,12	12.2729	3	23.562	6	10.3173	46.2062

5.1.2 Voice and Video Traffic Performance

The IEEE 802.11e standard prioritizes voice traffic over video by specifying a smaller contention window size for voice (Figure 5.1b). This means that video traffic gets blocked before voice when total traffic in the system approaches capacity. Through the mapping of social distance to wireless MAC AC's (5.1), SDA prioritizes real-time traffic first by the social distance, and then by traffic type. Hence, high priority (low social distance) flows of both voice and video traffic will have access to the channel when total traffic approaches capacity. We look at three cases of the traffic mix (of voice and video calls) to demonstrate the difference between IEEE 802.11e and SDA. In the following discussion NV denotes the number of voice calls, and NVI the number of video flows.

NV=NVI

Equal number of voice calls and video flows make up the real-time traffic in this case. Figure 5.2a shows the simulation results for this scenario. The x-axis shows the number of competing real-time flows in the network. The y-axis shows the number of voice and video flows accepted for both standard IEEE 802.11e and SDA.

For low traffic intensity (as compared to the system capacity), we see that both IEEE 802.11e and SDA perform identically. As the traffic intensity approaches capacity, SDA allocates resources (among voice and video flows) according to social distance rather than traffic type.

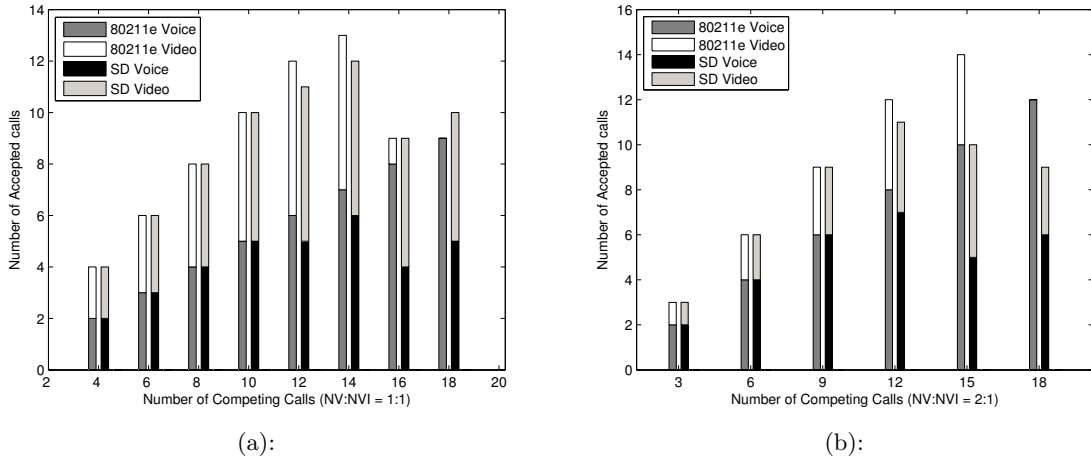


Figure 5.2: Network Performance for a mix of voice and video calls competing for the shared channel. Results for $NV=NVI$, $NV=2NVI$ and $2NV=NVI$ are shown in Figure 5.2a, Figure 5.2b, Figure 5.3a respectively. SDA performs better during saturation traffic conditions as compared to IEEE 802.11e.

As the number of calls exceeds 12, in the case of IEEE 802.11e we see that voice call traffic begins to overwhelm video flows. The worst case is when absolutely no video flows are able to access the network for the case of 18 (9 voice calls, 9 video flows) flows in the system. However, SDA still prioritizes by social distance, hence both voice and video flows in χ^1 are able to access the network. In SDA, the peak capacity of the system is reduced by one call, but the ratio of flows accessing the system is always closer to the incoming traffic profile (1:1) as compared to IEEE 802.11e. Also, SDA achieves resource distribution closer to the real-world priority of flows than IEEE 802.11e for traffic at system capacity.

The peak capacity of the system is reduced by one call in SDA due to the AC design. The call in VO(3) fails to meet the delay requirement and the mean opinion score falls below acceptable levels. In standard 802.11e however, this call is placed at VO and hence achieves desired quality. This can be alleviated by a different AC design than the one we have chosen here.

NV=2NVI

Figure 5.2b shows the simulation results for the scenario when the number of voice calls are twice the number of video flows in the system. The IEEE 802.11e MAC again prioritizes voice calls over video flows, with absolutely no video flows admitted when there are 18 flows in the system. SDA maintains the ratio of calls accepted close to the incoming traffic profile (2:1).

This is due to the allocation of resources by social distance rather than traffic type. Even at 18 calls, SDA accommodates 3 video calls as against none in the case of IEEE 802.11e. When the system is operating at capacity, both voice and video flows in χ^1 are allocated resources *before* voice (video) flows in χ^2 and so on.

2NV=NVI

The simulation results for the scenario where the number of video flows are twice the number of voice calls are shown in Figure 5.3a. When video calls outnumber voice calls by a 2:1 ratio, we can clearly see the unfairness of IEEE 802.11e in allocating resources. As the traffic approaches system capacity, voice calls can still starve the video flows out of resources. For calls exceeding 12 (4 voice calls) in the system, IEEE 802.11e barely accommodates any video flows at acceptable quality. SDA is able to serve both voice and video for every case, and it does not starve the voice flows in the process. All voice and video flows in χ^1 are allocated resources for every data point in the graph. SDA prioritizes flows by their social distance (χ^1 over χ^2), and then by the traffic type (VO_AC over VI_AC).

5.1.3 Discussion of Detailed Results

The detailed results for the scenario 2NV=NVI are shown in Table 5.1. For each scenario, we show the number of competing voice and video flows (NV,NVI), along with the number of them accepted into the system (# Calls, # Flows). For both voice and video traffic, we tabulate the aggregated mean opinion score for all the accepted flows. We compute the aggregate utility for best-effort HTTP flows too, followed by computation of the total network utility - which is the sum of utilities of voice, video and best-effort traffic.

As we can see from Table 5.1, IEEE 802.11e allocates all requested resources to voice flows first, followed by allocation to video flows, and the leftover capacity is used up by the best-effort traffic. As the number of voice calls increase, the total voice utility increases at the expense of video and best-effort traffic. IEEE 802.11e starts dropping video flows from 15 flows onwards, even though there are actually 10 video flows competing with 5 voice calls for resources. The packet loss ratio for video flows is too high, and almost none of the video flows achieve acceptable mean opinion score values.

SDA, on the other hand, allocates resources by the social distance. Therefore, with both voice calls and video flows competing for resources, all flows in χ^1 are allocated resources first, followed by flows in χ^2 and so on. The best-effort traffic uses up any leftover capacity. We see that beyond 12 flows in the system, video flows are still able to access the channel and achieve acceptable mean opinion scores. This happens at the expense of lower priority voice calls. We see that for high traffic intensities, voice flows in χ^2 do not achieve acceptable mean opinion

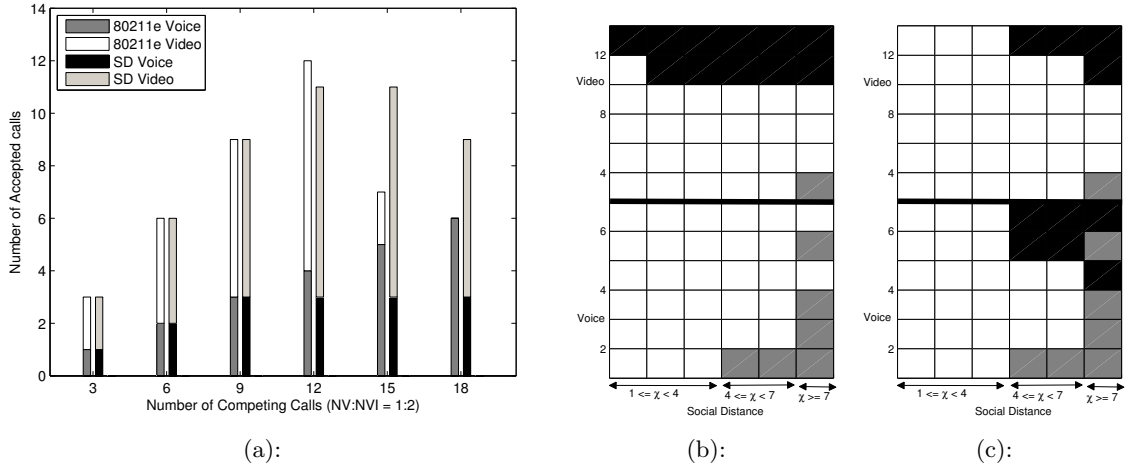


Figure 5.3: Specific resource allocation patterns for 5.3a are shown in 5.3b for IEEE 802.11e, and in 5.3c for SDA. White cells represent resources were allocated, black cells represent no resources were allocated, and grey cells represent no resources were requested. It can be seen that IEEE 802.11e allocates all requested resources to voice while starving video flows beyond 8 flows. SDA allocates resources to both voice and video flows in χ^1 before allocating resources to voice and video flows in χ^2 and so on.

score values. This happens due to the AC design in Figure 5.1b. Both voice calls and video flows in χ^2 are assigned to the same AC, and hence compete for the same resources. Eventually, even video flows in χ^2 cannot meet the quality requirements, but every flow (both voice and video) in χ^1 has access to the channel for every case shown in Table 5.1.

The total network utility achieved is higher in SDA for saturation traffic conditions as compared to IEEE 802.11e. We see that the peak utility (for a total of 12 flows) is higher for IEEE 802.11e, but it drops drastically with any increase in the number of voice/video flows. In case of SDA, the reduction in total utility is limited, and the total network utility achieved is always higher than IEEE 802.11e during saturation traffic conditions.

5.1.4 Resource Allocation Patterns

In order to further clarify the difference between IEEE 802.11e and SDA, we present a comparison of their resource allocation patterns for the scenario in Figure 5.3a. The resource allocation patterns are shown in Figure 5.3b and Figure 5.3c. In the patterns shown, we have one row of cells (rectangles) for each case (number of competing calls) in Figure 5.3a. The cells for voice and video flows are segregated, with the separator in the middle delineating the voice calls in the lower half from the video flows.

Each row is made up of 6 cells, with the first three cells representing flows in χ^1 , the next

two in χ^2 and the last cell represents flows in χ^3 . A white cell signifies that resources were allocated to the flows, a black cell signifies no resources were allocated, and a grey cell signifies there were no flows in that social distance class requesting resources.

Figure 5.3b shows the allocation through IEEE 802.11e for the scenario of 2NV=NVI. We see that the voice calls get all the requested resources for every scenario (1-6 calls). For the video flows, IEEE 802.11e is able to service up to 8 flows. For the case of 10 and 12 video flows, almost no resources are allocated to any video flow irrespective of the social distance. This would be an unfair allocation if the video flows were at a higher social distance than some of the voice flows in the system.

Figure 5.3c shows the allocation pattern for SDA for the same scenario. We see that video (and voice) flows in χ^1 are allocated resources for all cases (2-12 video flows, 1-6 voice calls). Video flows in χ^2 gain at the expense of voice calls in χ^2 because both of these share the same AC. Overall, there is a much more fairer allocation of resources across voice and video flows for all the cases considered. SDA allocates resources primarily along the dimension of social distance (horizontal in Figure 5.3b and Figure 5.3c) rather than along the traffic type (vertical in Figure 5.3b and Figure 5.3c).

5.2 Social Distance and Fairness

Social distance, as we have defined it, is always at the background of most real-time communication flows on the network. Resource allocation algorithms can benefit by being aware of the usually implicit social distance, and incorporate it into the microscopic flow-level decisions. Such a differentiation is bound to affect the degree of fairness (of the network) towards the flows accessing the network. However, when viewed along the social distance dimension, an equal allocation of resources during saturation traffic conditions would be considered as unfair. This is because all the flows lose resources equally, when some of the flows were at lower priority than the others.

This is true especially in the case of a wireless network due to the shared wireless channel. As shown in our theoretical as well as simulation results, equal distribution of resources results in unfair treatment of higher priority flows. This also results in a lowered total network utility at saturation traffic conditions. In such a scenario, fairness should indeed be measured by how differently the various priority flows are treated by the network. As we can see from our results, SDA performs much better than IEEE 802.11e in terms of service differentiation for higher priority flows. It also achieves a higher total network utility as compared to standard IEEE 802.11e.

5.3 Summary

Social distance can be used to measure the relative priority of a flow regardless of the traffic type. The corresponding social distance aware utility optimization problem (SDA) does not necessitate any change to the solution algorithm itself. However, in terms of implementation, more access categories may be needed to incorporate flows classified by social distance as well as by the traffic type. In a given QBSS, SDA allocates resources based on the social distance, as compared to IEEE 802.11e which allocates resources based on the traffic type. When the QBSS is functioning at capacity, both SDA and IEEE 802.11e block flows. Neither algorithm can attain the peak capacity in such a scenario, mainly due to the broadcast channel coupled with the backoff mechanism. The difference between SDA and IEEE 802.11e is regarding the choice of flows which are accepted into the QBSS. In the case of 802.11e, all traffic other than voice is eventually blocked from accessing the channel. This is due to the AC design which prioritizes voice traffic over everything else. SDA services all flows in χ^1 before allocating resources to flows in χ^2 and so on. Both voice and video flows in χ^1 are treated as high priority when compared to voice and video flows in χ^2 . Since the measured social distance represents the perceived real-world priority of the flow, high priority flows will always have access to the channel, irrespective of the traffic type.

Chapter 6

Social Distance Aware Content Distribution

The communication flows on the network can be broadly classified into two categories: end users communicating with each other using real-time flows (voice, video) and end users requesting content (text, video) hosted on the network. In the previous chapters we saw how to exploit the social graph structure in order to prioritize competing real-time flows between users. In this chapter, we will look at the problem of exploiting social graph structure of content in order to improve end user experience of the network. We specifically look at video content with inter-relationships (e.g., Youtube), which can also be categorized as multimedia content delivered over HTTP. The reason for this is that video content is invariant and thus a candidate for caching/distribution for faster access. In effect, content is being brought closer to the end user so as to improve the transfer times without changing anything in the user's access network.

The performance of such a caching system will depend on identifying the videos which should be cached and the appropriate duration of caching. We look at both of these questions from a social network perspective. Videos are inter-related through explicit relationships (related videos) which also determine, to an extent, the future access patterns of end users. To determine social distance between individual videos, we use the popularity of videos (number of views) as a measure of social distance. Specifically, we use the reciprocal of popularity as the edge weight for the relationship between two videos.

We propose that the decision to cache a video should be based on the combined popularity of the individual as well as related videos rather than simply based on individual popularity of a video. We identify timescales at which the inter-relationships between the videos can change through a longitudinal data set. Using the concepts of centrality of nodes, we rank the set of videos in the data set according to their perceived importance. In doing so, we compare three centrality techniques - degree, closeness and betweenness. We evaluate how these centralities

affect the performance of a cache. We show that “Closeness” centrality performs better than the other two in most cases. It performs at least as well as the other two in all cases. Finally, we show that a distributed cache mechanism employing the centrality method to rank videos can reduce the load on the network significantly for even moderate content cache sizes.

6.1 Motivation

A recent study [46] of traffic characteristics of the Internet found that HTTP dominates the traffic share accounting for 68% of the trace. Of this, around 34% traffic was found to be multimedia. With so much of the traffic on the Internet determined by HTTP-based multimedia, there exists a realistic possibility for caching videos, such as those from Youtube, to reduce traffic flows.

There have been several recent studies focusing on the characteristics of Youtube-like multimedia content. This includes video popularity analysis [22], usage patterns at the local level [23], category specific content request analysis etc. Popularity analysis of videos is done to analyze whether they follow the popular Zipf distribution similar to traditional web traffic. In particular [24] states that media access patterns on the Internet follow a stretched exponential distribution rather than a Zipf like distribution. In [22], the popularity analysis restricted to Youtube videos reveals that the distribution has a Zipf waist with a truncated tail similar to an exponential cutoff.

Our focus is on the design of a distributed cache, similar in principle to a Content Distribution Network (CDN) or a P2P distribution system. The objective is to select some of the most “valuable” videos and bring them closer to the clients requesting them so that redundant load on the network is reduced. We believe that the vast majority of such caching decisions will happen at head of the popularity curve as this is the region where the most popular videos (of the millions of videos of the data set) are present. Hence the Zipf properties should hold in the case of selecting videos for a content cache, since all these decisions happen where the popularity curve for media files follows Zipf behavior.

The design of a cache for local Youtube usage was studied in [50]. They infer that the local Youtube usage has no correlation with global popularity of videos. The decisions of caching were based mainly on the local popularity of individual videos. In this work, we are focused on the caching of videos at the *global level*. The emphasis here is to include the *relationships* between videos as a prime indicator of their value for caching. These relationships play an important part in determining end user access patterns. There is an inherent social structure between the videos and we believe that this should be included when making decisions related to content distribution and caching.

Our methodology is as follows. We first collect information on some of the globally popular

Youtube videos. Through this we can construct a network graph of the relationships between the videos themselves. We then apply social network analysis techniques to the resulting graph in order to identify the value of a video as compared to the others. Based on the ranking obtained, we then proceed to evaluate the performance of a caching mechanism selecting videos according to their ranks. In the process we compare three techniques of measuring the value of a video, and point out the suitability (or not) of each of the three techniques.

6.2 Data Collection

Youtube provides the Data API [51] which can be used to incorporate a lot of its functionalities in stand-alone applications. The use of the Data API is simplified through available library implementations in a number of programming languages (Java, Python etc). We used the Python based interface to implement our data collection application. We can collect information on several parameters for a given video entry using the API calls. Specifically we retrieve information about the number of views, average rating, number of raters and related videos for every candidate video entry. We collect information about the upload date and duration for every candidate video once separately since these are invariant. The candidate videos themselves are videos that belong to either of the following two sources: standard video feeds provided by Youtube or traces made publicly available from [47].

Some of the standard video feeds provided by Youtube are the “Top Rated” videos, “Most recent” or “Most viewed” videos among others. A complete list can be found at the website [51]. We use all the three feeds mentioned above as candidate sources. An API call querying one of the feeds, say Top Rated videos, returns a list of 25 videos in that feed. We compile a list of the related videos for two levels of depth for every video entry returned. For example, suppose candidateA is a video entry in the list of 25 videos returned. We find the set of related videos for candidateA, say {related1, related2,...}. We then find the related videos for related1, related2, etc. The complete list of all these videos along with their parameter information forms the output of a single data collection run. Each such run gives us information about more than 16,000 videos.

We use the “Science and Technology” category trace from [47] made available publicly at [25]. This trace contains information for more than 250,000 videos from this category. This information is more comprehensive than our traces for the standard feeds. We use this trace as a cross check, since this data has not been collected primarily for the relationships. The results that we obtain from this trace will serve as further proof in favor of our claim that the inter-relationships do play a role in determining the value of a video.

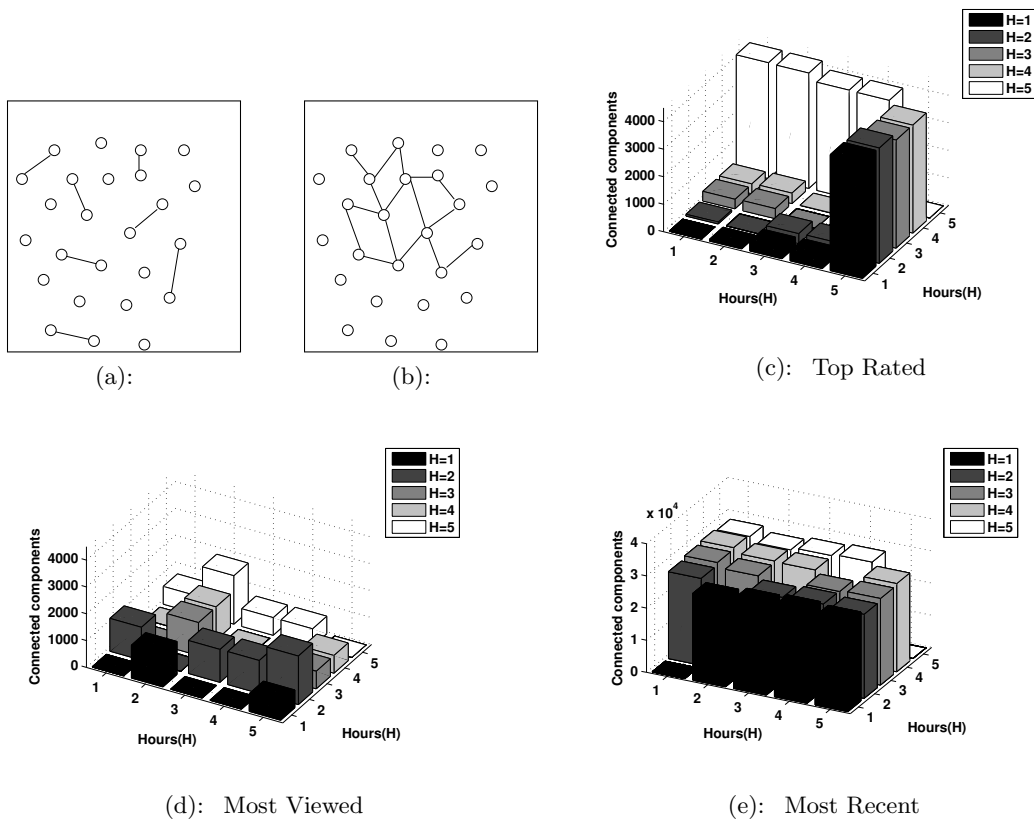


Figure 6.1: Structurally Figure 6.1b has more impact on design and performance decisions than Figure 6.1a, but both have the same ex-or counts. Variations in the inter-relationships are plotted against time for Top Rated (Figure 6.1c), Most Viewed (Figure 6.1d) and Most Recent (Figure 6.1e) videos.

6.3 Timescales

Here, we try to gain an understanding of the time scale at which the structure of the videos' inter-relationship changes significantly. This will be useful in identifying suitability for caching of a given source, as well as timescales at which the content cache will need to be updated. We conducted multiple data collection runs for each of the standard feeds for consecutive hours in a day. We did this for a total of 5 hours. We also gathered data on different days for 5 consecutive days. Through the Data API we can also collect information for the previous week, previous month and the previous year for the standard feeds. This gives us the longitudinal data set necessary to study the timescales for standard feeds.

As the reader might have realized, the output of a single data collection run is actually a graph. Our analysis is mainly looking at the difference between the graphs of the outputs. To

study the difference between any two graphs $G1$ and $G2$, we compute the Exclusive-OR of $G1$ and $G2$. Specifically,

$$G1 \oplus G2 = (V1 \cup V2, E1 \oplus E2)$$

This operation will result in a graph with zero or more connected components.

To quantify the difference between the two graphs, we compute the sum of the orders of all connected components. The order is given by the cardinality of the vertex set. This is just one of the ways of measuring the difference between the two graphs. It is simplistic in that the same numerical value is returned for the two very different results in Figure 6.1a and Figure 6.1b. The implication of these two structures will be vastly different for detailed design decisions. However, we think this measure is sufficiently detailed for our purposes of determining timescales of variations and suitability for caching.

We compute the graphical exclusive-or's of the longitudinal data that we gathered for the standard feeds and the results are shown in Figure 6.1. Figure 6.1c shows the variations across the hours for the Top Rated videos feed. The important feature of this set of captured data is that the structure changes significantly at the 5-hour mark. Note how the sum of connected components remains small when the first four hours of data are being compared with each other. However, the last data set has significant difference with all of the remaining four. This means that periodic cache updation may be necessary for this category of videos.

The variations for the Most Viewed videos feed, shown in Figure 6.1d, remain almost constant for the 5 hour period. However, the average is higher than that of the Top Rated feed. This points to the content cache size and its importance in the case of videos belonging to this category.

The Most Recent videos feed (Figure 6.1e) shows the largest variations of all the three. Intuitively, this should be the case for the first set of 25 videos. However, we expected the two levels of related videos to have something in common. This feed was selected since new videos are considered to be the ones visited the most. As we can see from the graph, this feed is not suitable for caching even on an hourly time scale.

6.4 Centrality

The most important decision in Youtube-like media content caching is the selection of videos to cache. In classical cache design, the concept of locality of reference says that the very fact of a variable being referenced is enough to place it in the cache. In case of multimedia content which is socially inter-related, there are additional parameters (popularity, ranking) associated with every video which make them different from each other. The process of deciding which

video to cache has to take these parameters into account. We think that in addition to the parameters, the structure (related videos) also has to be considered when designing content caches or deciding on content distribution throughout the network.

We need a method of mathematically quantifying the position/value of a node in a network. The metrics for this have been proposed in Social Network Analysis (SNA) techniques. We use one such important concept from SNA and try to analyze how beneficial the structure (taken in combination with the parameters) will be in arriving at a content caching decision.

The importance of a node in a social network graph is determined in terms of its **centrality**. Several methods have been proposed to measure the centrality of a node. We will use and compare three different methods to rank the nodes of the graph by their centrality, namely, degree, closeness and betweenness.

6.4.1 Degree Centrality

As the name suggests, the degree of a node determines the rank of the node in this definition of centrality. In case of a directed graph it is the out degree of the node. For the case of the standard feeds, the degree of every node is almost always constant at 25 because of the way we have collected the data. Thus, for the standard feeds, degree centrality will place most videos at the same rank.

6.4.2 Closeness Centrality

The closeness centrality of a node is a measure of how quickly one can reach all the connected nodes. The edge weights determine the cost of traversing a link to an adjacent node. We set the weight of an edge as follows:

$$weight_{(v_1 \rightarrow v_2)} = \frac{1}{views_{v_2}}$$

where $views_{v_2}$ is the number of views recorded for the video v_2 . The closeness centrality is computed using the following formula [10]:

$$C_C(v_i) = \frac{N - 1}{\sum_{j=1}^N w(v_i, v_j)} \tag{6.1}$$

where N is the number of reachable nodes from v_i and $w(v_i, v_j)$ is the weight of the shortest path from v_i to v_j . The nodes which are related to videos with the highest views will be ranked higher according to this measure.

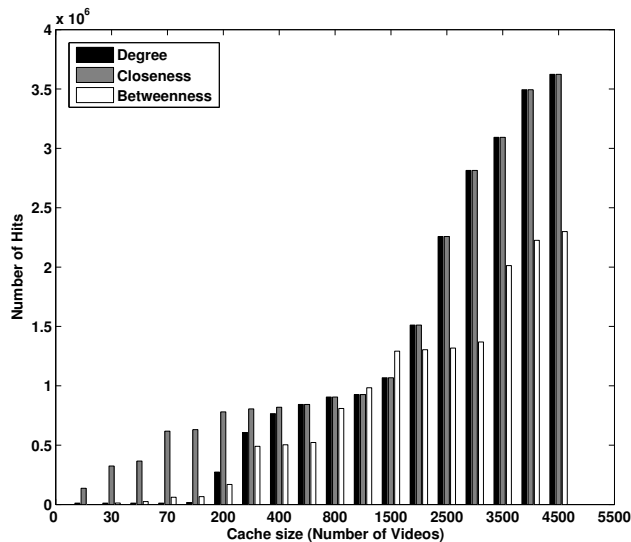


Figure 6.2: Comparison of the three centrality methods for the Top Rated videos feed. Betweenness centrality saturates earlier than the other two, it also performs poorly at choosing the best videos as compared to closeness centrality. Also, at small content cache sizes, closeness centrality performs exceptionally well than the other two.

6.4.3 Betweenness Centrality

Betweenness centrality is defined to be a measure of how involved a node is in the shortest paths in the network. Specifically, for a given node, betweenness centrality is the count of the number of shortest/geodesic paths which traverse through the node. If P_{ij} is the number of geodesic paths from i to j , and P_{ikj} of these pass through the node k , then the betweenness centrality of node k is given by:

$$\sum_i \sum_j \frac{P_{ikj}}{P_{ij}}, \quad i \neq j \neq k \tag{6.2}$$

These are three different ways of measuring the value of a node. We now compare these three centrality schemes based on which of them performs the best at identifying the most valuable videos.

6.5 Performance Analysis

In order to evaluate how the different measures of centrality affect the content cache performance, we rank the nodes according to their centrality (degree, closeness or betweenness). The

content cache chooses videos according to their rank and limited by the cache size measured in number of videos. From the longitudinal data that we have collected, we construct a working set of the future accesses for all the videos in the data set. We can now measure the number of hits, and also compare the three centrality techniques to see which of them can better identify the most valuable videos.

6.5.1 Standard Feeds

The performance of the cache for Top Rated standard feed can be seen in Figure 6.2. From Figure 6.2 we can see the number of hits in the content cache for all the three centrality schemes, and for varying cache sizes. There are several interesting things to note here. Firstly, for small content cache sizes, from 10 and up to 200 videos, Closeness centrality far outperforms the other two. The ranking according to Closeness identifies the set of valuable videos better than the other two. Note also that at large content cache sizes degree centrality catches up with closeness. Degree centrality performs worse at small sizes because of the fact that many nodes have the same degree, thereby all of them being flat ranked into a single category. In our results we see that every node has a unique Closeness centrality measure.

Betweenness centrality in Figure 6.2 tends to saturate earlier than the other two techniques. It cannot achieve the maximums that Closeness centrality is able to achieve. This may be due to the fact that these results are for the Top Rated Videos standard feed. Because of the way we are collecting the data, it is not a complete snapshot of the Youtube video graph, but rather a snapshot of a subgraph. Betweenness may work better if the entire graph information is available. We go into more detail to see why the difference between the Closeness and Betweenness centralities exists.

Figure 6.3a and Figure 6.3b show how Closeness and Betweenness make their video choices for a content cache size of 5000 videos. The working set (set of videos accessed) contains ~ 1600 videos. The figures show us how well the two centrality techniques can select the most accessed videos. Each square in the figure represents 1% of the videos accessed (~ 16). Furthermore, some videos of the working set get more views than the others. The bottom-left square in both these graphs represents the top 1% of the views. We move left-to-right and bottom-to-top, and the videos with the least 1% of views are grouped together at the top-right square. Gray and Black squares mean less videos were chosen from this group. As we can see, Closeness centrality does a much better job overall of choosing videos. Betweenness performs poorly in almost every group. Note that this is for the Top Rated Videos standard feed.

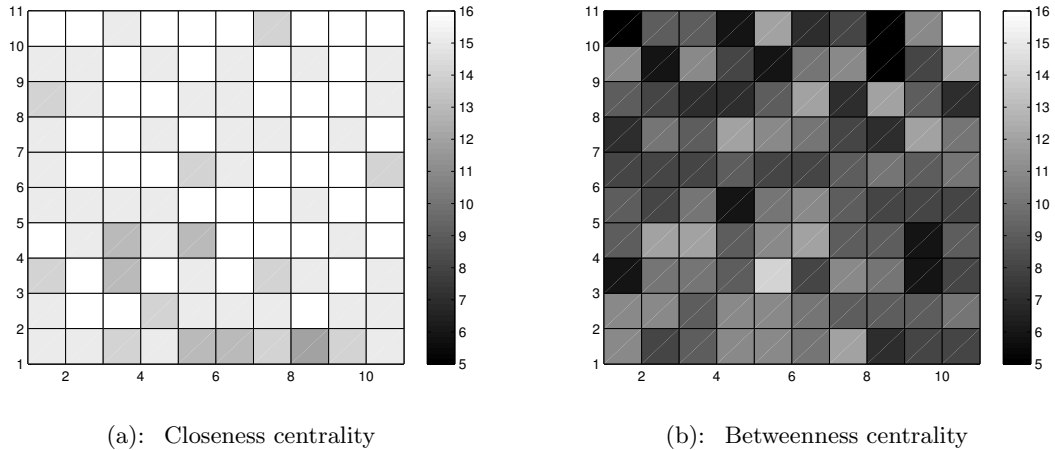


Figure 6.3: Figure 6.3a shows the number of hits for cached videos ranked by closeness centrality. Each cell (square) represents 16 videos, with the lower-left cell representing the most viewed and the upper-right representing the least viewed videos. Figure 6.3b shows that betweenness centrality performs poorly in choosing the right videos, since the number of hits is low across the board.

6.5.2 Science and Technology Category Trace

The plot for the number of hits in this category is shown in Figure 6.4a. This is a much more detailed trace than the standard feeds and has more of the complete graph information. We see that Betweenness centrality does not tend to saturate in this case though it still does not perform as well as Closeness. At low content cache sizes, Closeness centrality still trumps both Degree and Betweenness. Hence we infer that Closeness centrality is the best of the three in identifying the most valuable videos.

6.5.3 Number of videos served

With a content cache size of 2500 videos, we can see from Figure 6.2 that more than 2 million requests for the Top Rated videos could be served from the content cache. For the Science and Technology trace the average number of hits for video requests in the content cache is around 100,000. In the absence of a content cache all these requests will be directed to the content servers.

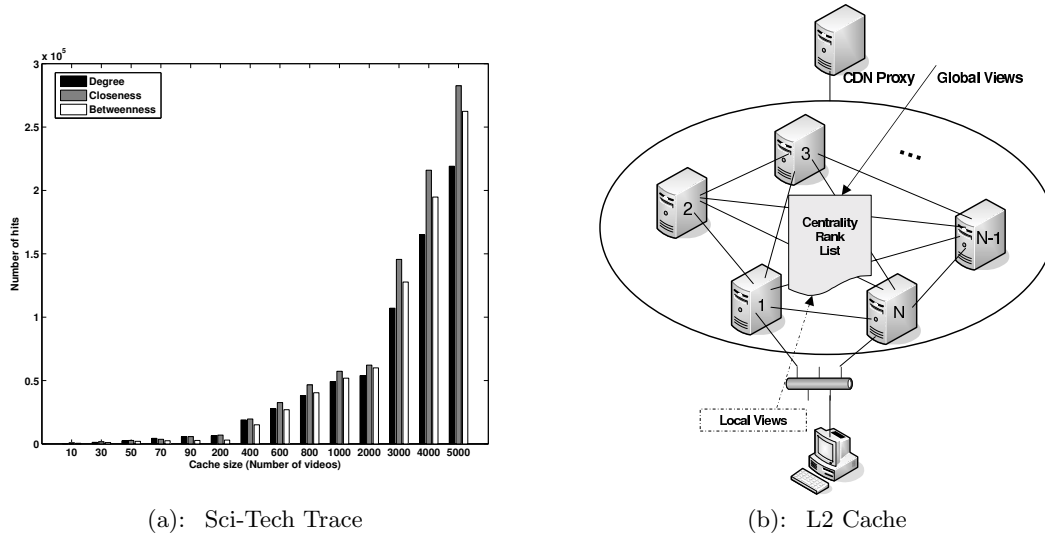


Figure 6.4: The performance of cache for Science and Technology trace is shown in Figure 6.4a. Figure 6.4b shows a proposed L2 distributed content cache.

6.6 Proposed L2 Distributed Content Cache

An actual implementation of a second level distributed content cache may look as shown in Figure 6.4b. This is modeled on the *structured P2P* distributed hash-table based system [52]. The L2 content cache has a total of N servers with each server uniquely labeled. Addition of new servers is thus very simple in this case. The overlay network for the L2 servers is assumed to be a fully connected graph.

The location of the videos in the distributed cache is deterministic. Given a total of M videos, each server will hold an equal fraction of the videos $\frac{M}{N}$. A video with rank k according to the centrality scheme will be cached at the server $\lfloor \frac{kN}{M} \rfloor + 1$. Such a fixed video distribution scheme ensures a $O(1)$ lookup time in the content cache.

The centrality list may be computed from the global views of the multimedia data source. It can also be based on the local viewing patterns of an organization or autonomous system. In fact, there may be two concurrent lists for both the global and local viewing patterns, and the content cache can be shared equally between these two lists of videos. Our results are applicable for the global viewing patterns of Youtube-like videos. But we emphasize that the proposed L2 content cache is only dependent on the centrality based rank list, and not on the source of this rank information.

6.7 Summary

Social inter-relationships between content can predict the future access patterns of users. We look at the social graph of videos, which are invariant and hence suitable for caching, to determine feasibility of bringing content closer to the user. We also proposed a methodology to study the timescales for changes in inter-relationships of Youtube-like videos in general.

The social distance between content items is chosen to be inversely dependent on the popularity of the video. Thus, a highly popular video will have a low cost from its neighboring (related) videos. Using the resulting social network structure, we perform centrality analysis to determine the most important videos in the social graph. We used three different centrality methods to quantify the value of a video and compared their performance over different data sets. We conclude that Closeness centrality performs close to the best of the three at all content cache sizes in both the scenarios. Hence it is the best measure to use for ranking videos. We believe this can be used in improving the design and performance of distributed caching solutions, content distribution systems. We provide one possible design of an actual L2 distributed content cache based on the centrality scheme of ranking.

Chapter 7

Experimental Data Trace Collection for SDA

Implementation of SDA in a real-world scenario requires identification of the underlying social network, along with the perceived social distances. We create this scenario for our campus using specific applications written for mobile phones, and use these applications to capture data traces regarding user interactions and experience (QoS). In addition to capturing the flow level interactions between users, we also log information about their physical locations in the building. We achieve this by implementing our own WiFi-based localization algorithm which can place users inside the building in the absence of GPS. Users are involved in group activities using static itineraries communicated through a Longbow server to the client applications. We monitor statistics of usage such as end-to-end delay, packet loss, data transferred and so on, and log all this information. This can be used to construct a social distance aware utility function for our mobile phone application. If the access point can be made aware of the social structure, it can differentiate between competing flows depending on the social distance as well as the traffic type.

7.1 WiFi Localization

Physical location of users can influence their communication patterns, and an explicit knowledge of the location can be used to predict a user's resource requirements from the network. As a first step, it is necessary to include location information along with the traffic traces to understand if there are any correlations between the two. At the same time, location-dependent applications have been implemented which present a different interface to the end user based on the physical location. For both these cases, it is thus necessary to locate the user accurately.

We consider the case of portable hand-held devices being used by the end users to com-

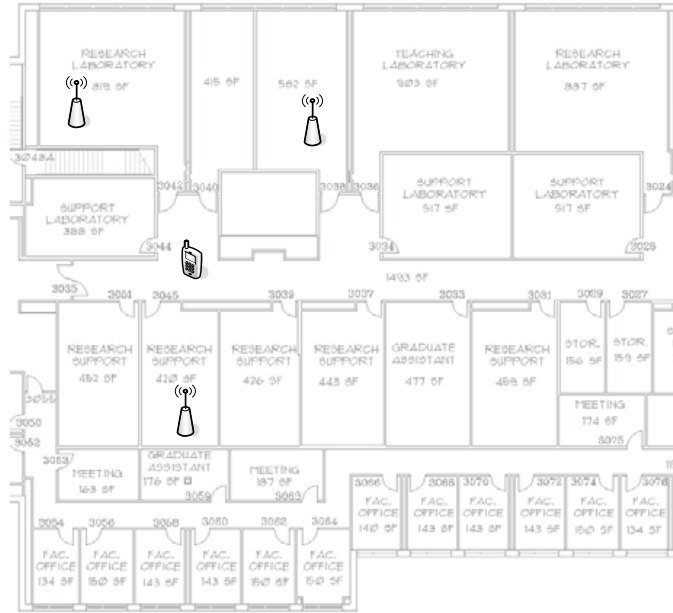


Figure 7.1: Creating a wireless signal map of the building requires capturing RSSI information at different locations in the building.

communicate with each other or request content from the network. In such devices, the location of a user can be ascertained through GPS (most mobile phones today are GPS-capable) when they are located outdoors. However, inside a building the phones cannot get a lock on the GPS satellite(s) and thus localization is not possible using GPS indoors. To overcome this, we implement our own wireless signal strength based localization for users situated inside a building.

7.1.1 Generation of the wireless map

The idea of WiFi localization is to create a wireless signal map of the building through off-line measurements and use it for locating end-users during the online phase. Such a localization scheme will only work for buildings which have a wireless local area network installed. The off-line phase involves accumulating measurements taken at a few chosen locations in the building, followed by generation of a finer-grained map using filtering techniques.

We used the actual end-user devices themselves to capture the wireless signal measurements (Figure 7.1). The data which is recorded for each chosen location consists of tuples, where each tuple is made up of a MAC address (of a wireless access point) and the corresponding received signal strength (RSSI). The RSSI is usually expressed in dbM, but there is no standard way of measuring RSSI in the IEEE standard. Thus, each device implements its own version of

computing RSSI. This is the reason why we used the target end-user devices to take the off-line measurements too.

The collected data regarding RSSI signatures for various locations in the building is saved in a database. The signal strength fingerprint of a location is associated with the corresponding location co-ordinates, which are expressed in (Latitude, Longitude) format. To achieve this with the target end-user devices, we implemented a stand-alone application for the device which presents the user with overlays for each floor of the building. When the user taps on any location on the floor overlay, the application converts the screen co-ordinates to (Latitude, Longitude) format. This is done using an *affine transform*, which is determined by performing linear regression fitting for a few landmarks in the building (correlating screen co-ordinates for these landmarks with their corresponding latitude,longitude values). Following this, the application measures the wireless signal strength information and relays the combined (Location,RSSI) information to our location server.

The procedure to capture RSSI-signatures is repeated at several locations in the building (multiple points on each floor, one point for each room and so on). All of this information is stored in a database. The localization algorithm accesses this database to determine location of an end-user device during the online phase of WiFi localization.

7.1.2 Localization of a user

When a device requests for positioning information, it sends to our location-server the RSSI-signature for its current physical location. The server (implemented in Python) then accesses the database to determine the closest match to the received signal strength signature from among the measurements which are present in the database. The closest match is implemented as a range matching algorithm, with all tuples matching within a given range (of the measured RSSI) being identified as possible candidates for the location of the end-user.

Upon identifying candidate physical locations in the building, we then evaluate a majority function on the candidates to determine the candidate location with the highest probability (of the mobile device being there). This location is returned to the client device, which then displays it on the user interface. This process is repeated every 5 seconds for each device. The most recent location of a user is saved at the location server, and can be provided to relationship peers of this user when they request for it.

With the location of an end-user determined inside the building, we now implement phone-based applications which augment this functionality with group activities or scenarios involving group communication over the network. This is done in order to demonstrate the feasibility of social distance aware resource allocation when communication patterns explicitly involve group communication over the network.

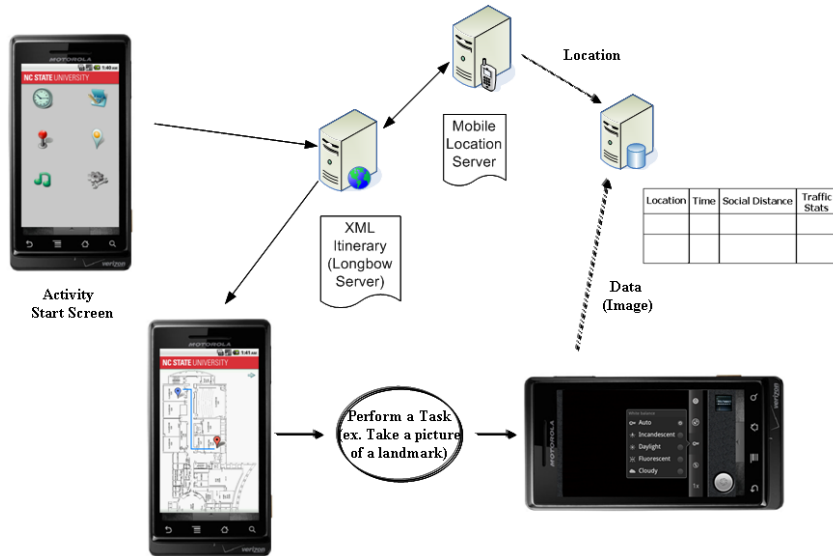


Figure 7.2: Sequence of messages exchanged to achieve the desired objective as determined by the itinerary server.

7.2 Client Applications

We designed two separate client applications for different end-user devices (iPhone and Android). The objectives of both these applications was to provide the end-user with a way to declare the social relationships and the corresponding relationship weights. Since the client applications augment WiFi localization, the ability to locate a user is already in-built. The applications provide capabilities to display locations of all relationship peers within the building.

Following the identification of social groups and relationships by the end user, the client application has the capability to instantiate group activities between related users. The activities may involve actions like going to a given location in the building, answering questions regarding that location, taking pictures of landmarks at the location and so on. While the users are doing all of this through the wireless network, our client application logs information about the current state of the social network, the achieved network performance for the group activity (QoS) etc.

The activities are statically coded and stored in XML format on a Longbow server. The reason for doing this is that, eventually, the Longbow server may be used to deliver adaptable itineraries. The activity changes dynamically based on the choices that the end-users make. This can be achieved through Longbow, but for our work, we implement static itineraries stored on the Longbow server.

Another key feature of the client applications is the ability to display paths between different

locations in the building. This is implemented using the PathFinder module, which takes a weighted geographical graph of the building as input along with source and destination points, and generates a shortest path route to get to the destination from the source location. The weighted graph of geographical points in the building was generated off-line for each floor of the building.

A sequence of events for the Android application can be seen in Figure 7.2.

Chapter 8

Conclusion

The flows initiated by end users over the network can be broadly classified as either being a real-time communication between two end-users, or a request for transfer of content hosted on the network. In both cases, the mode of communication (voice, video, text) determines a desired service differentiation for this flow with respect to other competing flows. The desired level of service is expressed in terms of quality of service (QoS) requirements from the network. However, the quality of service (QoS) differentiation in networks has been traditionally designed to be attending to *only* the traffic type of the requesting flow. Since the individuals communicating over the network are human end users, there is an underlying social context which drives the communication requests. When the network is agnostic to this important attribute of end-user communication, it forgoes the ability to provide a finer distinction between the quality of service delivered to competing flows.

We propose a way to measure the social context associated with communication flows using the concept of social distance. The end users of a shared communication network identify their relationship peers in the wider social network. They also provide a measure of how important the relationship is to them, when compared to all of their other relationships. This is the social distance, as chosen by the end user based on the importance of a relationship to them *personally*. Using this, we create a global social-network-wide social distance measure by combining the user's preference with their overall importance in the wider social network. This is done by measuring the centrality of users in the social network. This was discussed in detail in Chapter 2, where we also gave an example social network for which this process was applied to generate the social distance matrix.

Following the measurement of social distance between users, we need to incorporate it into network resource allocation decisions in order to provide better service differentiation among competing flows. We do this by introducing utility bounds on flows depending on their social distance. Thus, a flow with a higher social distance (lower priority) will have a lower maximum

utility bound than a flow with lower social distance. Through such bounds, the network can differentiate between flows of the same traffic type, but different social distance. We formulated the generic social distance aware utility function in Chapter 2, and we also provide an example for the case of elastic traffic in the network.

In our work, we focus on resource allocation decisions in a wireless network mainly due to the resource constrained medium which makes resource allocation particularly important, but also due to the ability to control allocation using discrete parameters at the wireless MAC layer. In Chapter 4, we discuss social distance aware resource allocation (SDA) for voice flows in a QoS capable wireless network (IEEE 802.11e QBSS). We show, through theoretical analysis as well as simulation results, that SDA increases the total network utility achieved for every case considered when compared to IEEE 802.11e. Following this, in Chapter 5 we consider the case of both voice and video flows competing for shared resources over an IEEE 802.11e QBSS. Our simulation results again show that the total network utility is increased by SDA as compared to standard IEEE 802.11e. This happens because the resource allocation decisions are determined by looking at the dimension of social distance in the case of SDA, rather than solely based on the traffic type as in the case of IEEE 802.11e.

Following this, we look at the case of communication flows which involve content transfer from the network. We propose that relationships between content can be used to determine the relative importance of individual content within the wider content-space. In the case where such relationships are explicit, such as social content (Youtube), the social network is already determined. The weights of the relationship edges are determined in our work by measuring the popularity of content and maintaining the weights of edges going to popular content (in the social network) at a lower distance. In Chapter 6, we use centrality measures on such a social graph for content to determine the most relevant data and cache it using a distributed caching mechanism. We provide performance results about how many requests for content were satisfied through the cache for the different centrality measures. We conclude that closeness centrality is the best measure for determining the most relevant content from analyzing the social content graph.

Finally, we provide details of some mobile phone applications that we implemented to measure the effect of social networks and social distance on the resulting communication. The important components of these applications are the WiFi localization module, the itinerary generation module and the PathFinder module. Working in tandem, these modules create scenarios where a group of users communicate due to certain underlying social context determined by the itinerary server. We measure the traffic profile generated, the network performance for duration of the group activity, location information for the users and store all of this information in a central database.

REFERENCES

- [1] D. Awduche, A. Chiu, A. Elwalid, I. Widjaja, and I. Widjaja. RFC 3272: Overview and Principles of Internet Traffic Engineering. 2002.
- [2] Jiayue He, Jennifer Rexford, and Mung Chiang. Don't optimize existing protocols, design optimizable protocols. *SIGCOMM Comput. Commun. Rev.*, 37:53–58, Jul. 2007.
- [3] Anders Gunnar, Mikael Johansson, and Thomas Telkamp. Traffic matrix estimation on a large IP backbone: a comparison on real data. In *Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*, IMC '04, pages 149–160. ACM, 2004.
- [4] Munmun De Choudhury, Hari Sundaram, Ajita John, and Doree Seligmann. Dynamic Prediction of Communication Flow Using Social Context. In *Proceedings of the nineteenth ACM conference on Hypertext and hypermedia*, HT '08, pages 49–54, 2008.
- [5] Yongxiang Xia, Chi K. Tse, Francis C.M. Lau, Wai Man Tam, and Michael Small. Analysis of telephone network traffic based on a complex user network. *Physica A: Statistical Mechanics and its Applications*, 368(2):583–594, 2006.
- [6] Ryan Rowe, German Creamer, Shlomo Hershkop, and Salvatore J Stolfo. Automated social hierarchy detection through email network analysis. In *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis*, pages 109–117. ACM, 2007.
- [7] Kevin Lewis, Jason Kaufman, Marco Gonzalez, Andreas Wimmer, and Nicholas Christakis. Tastes, ties, and time: A new social network dataset using facebook.com. *Social Networks*, 30(4):330 – 342, 2008.
- [8] P.O. Boykin and V.P. Roychowdhury. Leveraging social networks to fight spam. *Computer*, 38(4):61–68, Apr. 2005.
- [9] R. Kumar, P. Ragbavan, S. Rajagopalan, and A. Tomkins. The Web and social networks. *Computer*, 35(11):32–36, Nov. 2002.
- [10] E. M. Daly and M. Haahr. Social network analysis for information flow in disconnected delay-tolerant MANETs. *Mobile Computing, IEEE Transactions on*, 8(5):606–621, 2009.
- [11] Emory S. Bogardus. *A Social Distance Scale*, 1933. http://www.brocku.ca/MeadProject/Bogardus/Bogardus_1933.html.
- [12] Emory S. Bogardus. *Measuring Social Distance*, 1925. http://www.brocku.ca/MeadProject/Bogardus/Bogardus_1925c.html.
- [13] Munmun De Choudhury, Winter A. Mason, Jake M. Hofman, and Duncan J. Watts. Inferring relevant social networks from interpersonal communication. In *Proceedings of the 19th international conference on World wide web*, WWW '10, pages 301–310. ACM, April 2010.

- [14] Stephen P. Borgatti. Centrality and Network Flow. *Social Networks*, 27(1):55–71, Jan. 2005.
- [15] L. Freeman. Centrality in valued graphs: A measure of betweenness based on network flow. *Social Networks*, 13(2):141–154, Jun. 1991.
- [16] Linton C. Freeman. A Set of Measures of Centrality Based on Betweenness. *Sociometry*, 40(1):35–41, Mar. 1977.
- [17] Phillip Bonacich. Power and Centrality: A Family of Measures. *American Journal of Sociology*, 92(5):1170–1182, 1987.
- [18] Phillip Bonacich. Some unique properties of eigenvector centrality. *Social Networks*, 29(4):555–564, 2007.
- [19] Mung Chiang, S.H. Low, A.R. Calderbank, and J.C. Doyle. Layering as Optimization Decomposition: A Mathematical Theory of Network Architectures. *Proceedings of the IEEE*, 95(1):255–312, Jan. 2007.
- [20] Rayadurgam Srikant. *The Mathematics of Internet Congestion Control (Systems and Control: Foundations and Applications)*. SpringerVerlag, 2004.
- [21] S.H. Low. A duality model of TCP and queue management algorithms. *IEEE/ACM Trans. Netw.*, 11(4):525–536, Aug. 2003.
- [22] Meeyoung Cha, Haewoon Kwak, Pablo Rodriguez, Yong-Yeol Ahn, and Sue Moon. I Tube, You Tube, Everybody Tubes: Analyzing the World’s Largest User Generated Content Video System. In *ACM Internet Measurement Conference*, October 2007.
- [23] Phillipa Gill, Martin Arlitt, Zongpeng Li, and Anirban Mahanti. Youtube Traffic Characterization: A View From the Edge. In *IMC ’07: Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, pages 15–28. ACM, 2007.
- [24] Lei Guo, Enhua Tan, Songqing Chen, Zhen Xiao, and Xiaodong Zhang. The Stretched Exponential Distribution of Internet Media Access Patterns. In *PODC ’08: Proceedings of the twenty-seventh ACM symposium on Principles of distributed computing*, pages 283–294. ACM, 2008.
- [25] Youtube Traces. <http://an.kaist.ac.kr/traces/IMC2007.html>.
- [26] S. Ganguly, V. Navda, K. Kim, A. Kashyap, D. Niculescu, R. Izmailov, S. Hong, and S. Das. Performance Optimizations for Deploying VoIP Services in Mesh Networks. *IEEE Journal on Selected Areas in Communication*, 24(11):2147–2158, November 2006.
- [27] Hung yu Wei, Kyungtae Kim, Anand Kashyap, and Samrat Ganguly. On Admission of VoIP Calls Over Wireless Mesh Network. *IEEE International Conference on Communications*, 5:1990–1995, June 2006.
- [28] Xudong Wang, Abhishek Patil, and Weiling Wang. VoIP over Wireless Mesh Networks: Challenges and Approaches. *ACM International Conference Proceedings*, 2006.

- [29] Parag S. Mogre, Matthias Hollick, and Ralf Steinmetz. QoS in Wireless Mesh Networks: Challenges, Pitfalls, and Roadmap to its Realization. *Network and Operating System Support for Digital Audio and Video (NOSSDAV 2007)*, 2007.
- [30] Jie Hui and Michael Devetsikiotis. Metamodeling of Wi-Fi Performance. *IEEE International Conference on Communications*, 2:527–534, June 2006.
- [31] Fabrizio Granelli and Michael Devetsikiotis. Designing Cross-Layering Solutions for Wireless Networks: a General Framework and Its Application to a Voice-over-WiFi Scenario. *11th International Workshop on Computer-Aided Modeling and Design of Communication Links and Networks (CAMAD'06)*, pages 1–7, June 2006.
- [32] ITU-T Recommendation G.711. *Pulse code modulation (PCM) of voice frequencies*, 1988. <http://www.itu.int/rec/T-REC-G.711/>.
- [33] ITU-T Recommendation G.107. *The E-model: a computational model for use in transmission planning*, 1988. <http://www.itu.int/rec/T-REC-G.711/>.
- [34] S. Sengupta, M. Chatterjee, and S. Ganguly. Improving quality of voip streams over wimax. *IEEE Trans. Comput.*, 57(2):145–156, Feb. 2008.
- [35] D.P. Palomar and Mung Chiang. A tutorial on decomposition methods for network utility maximization. *IEEE J. Sel. Areas Commun.*, 24(8):1439–1451, Aug. 2006.
- [36] M. Chiang, Chee Wei Tan, D.P. Palomar, D. O’Neill, and D. Julian. Power Control By Geometric Programming. *IEEE Trans. Wireless Commun.*, 6(7):2640–2651, Jul. 2007.
- [37] I. Inan, F. Keceli, and E. Ayanoglu. A capacity analysis framework for the IEEE 802.11e contention-based infrastructure basic service set. *IEEE Trans. Commun.*, 57(11):3433–3445, Nov. 2009.
- [38] J.W. Robinson and T.S. Randhawa. Saturation throughput analysis of IEEE 802.11e enhanced distributed coordination function. *IEEE J. Sel. Areas Commun.*, 22(5):917–928, Jun. 2004.
- [39] Albert Banchs and Luca Vulliamy. Throughput analysis and optimal configuration of 802.11e EDCA. *Comput. Netw.*, 50:1749–1768, Aug. 2006.
- [40] Zhen ning Kong, D.H.K. Tsang, B. Bensaou, and Deyun Gao. Performance analysis of IEEE 802.11e contention-based channel access. *IEEE J. Sel. Areas Commun.*, 22(10):2095–2106, Dec. 2004.
- [41] Dongxia Xu, T. Sakurai, and H.L. Vu. An access delay model for IEEE 802.11e EDCA. *IEEE Trans. Mobile Comput.*, 8(2):261–275, Feb. 2009.
- [42] Dragos Niculescu, Samrat Ganguly, Kyungtae Kim, and Rauf Izmailov. Performance of VoIP in a 802.11 Wireless Mesh Network. *INFOCOMM 2006 - 25th IEEE International Conference on Computer Communications*, pages 1–11, Apr. 2006.

- [43] Jin Cao, W.S. Cleveland, Yuan Gao, K. Jeffay, F.D. Smith, and M. Weigle. Stochastic models for generating synthetic HTTP source traffic. *INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies*, 3:1546–1557 vol.3, March 2004.
- [44] Mirco Musolesi and Cecilia Mascolo. Designing mobility models based on social network theory. *SIGMOBILE Mob. Comput. Commun. Rev.*, 11(3):59–70, 2007.
- [45] *Video Trace Library*. <http://trace.eas.asu.edu/yuv/index.html>.
- [46] Jeffrey Erman, Alexandre Gerber, Mohammad Taghi Hajiaghayi, Dan Pei, and Oliver Spatscheck. Network-Aware Forward Caching. In *WWW*, pages 291–300. ACM, 2009.
- [47] M. Zink, K. Suh, Y. Gu, and J. Kurose. Watch Global, Cache Local: YouTube network traffic at a campus network: measurements and implications. In *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 6818 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, January 2008.
- [48] Data API Protocol Reference Guide. <http://code.google.com/apis/youtube/1.0/reference.html>.
- [49] Meeyoung Cha, Haewoon Kwak, Pablo Rodriguez, Yong-Yeol Ahn, and Sue Moon. I Tube, You Tube, Everybody Tubes: Analyzing the World’s Largest User Generated Content Video System. In *ACM Internet Measurement Conference*, October 2007.
- [50] Keong Lua, J. Crowcroft, M. Pias, R. Sharma, and S. Lim. A Survey and Comparison of Peer-to-Peer Overlay Network Schemes. *Communications Surveys & Tutorials, IEEE*, pages 72–93, 2005.