Design of Multi-tier Wireless Mesh Networks

Raghuraman Rangarajan

Advisor
Prof. Sridhar Iyer
-
July 2009, IIT Bombay

Goal

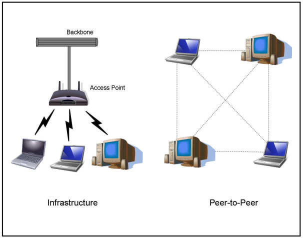
Design wireless data networks

- mesh networks and wireless local area networks
- capacity constraints

Purpose

- Construct topology
- Position infrastructure nodes
- Provision bandwidth

Overview of Wireless Networks



Wireless data networks can be used as

- Infrastructure or peer-to-peer (802.11)
- Local (WLAN) or Backhaul networks (802.16, Mesh)

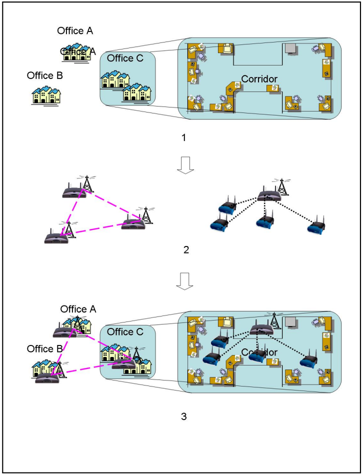
Wireless vs Wired

- Removes physical connectivity
- Allows user mobility
- Re-configuration of network incurs minimal cost
- Wired n/w have higher data rates
- Capacity provisioning important in wireless n/ws

Outline

Wireless Network Design I

Example Campus Network

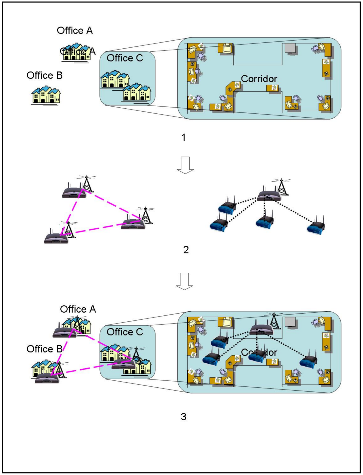


Network elements

- User devices
- Last-hop access (APs)
- Backhaul network (Routers)
- Application services

Wireless Network Design II

Design problem



Construct network topology satisfying design constraints

Design constraints

- Coverage
- Capacity
- Application scenarios
- Heterogeneous technologies
- Cost

Wireless Network Design III

Generic Design Problem

Network design problem (NDP)

Given client nodes and deployment layout
Construct network topology
Subject to constraints
While minimizing network infrastructure cost

Wireless Network Design IV

Current approaches

- Site survey
- Simulations
- Test measurements
- Signal strength measurements
- RF planning

Drawbacks [Mclean, How to design a WLAN, 2003]

- Difficult to provision 802.11 DCF
- Suitable for small-sized networks
- Address only coverage issues

Need

Integrated Approach to Wireless Design

Wireless Network Design V

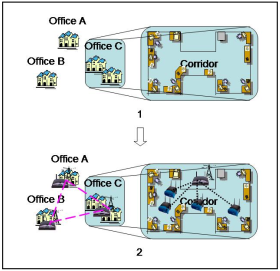
Design issues

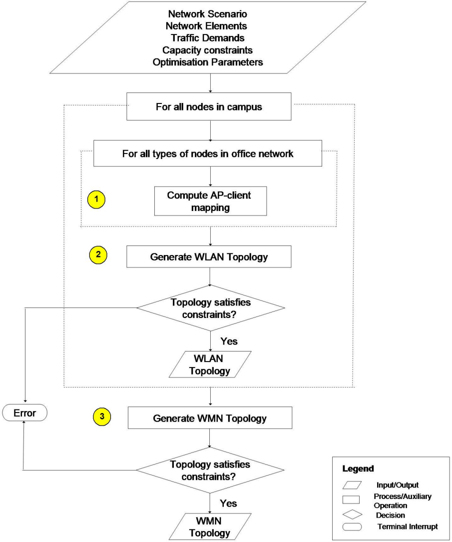
- Provisioning 802.11 WLANs in heterogeneous application scenarios
- Capacity-constrained wireless network design
- Minimising network infrastructure cost
- Integrated design of local area and backhaul wireless networks

Outline

Bottom-up Design Flowchart

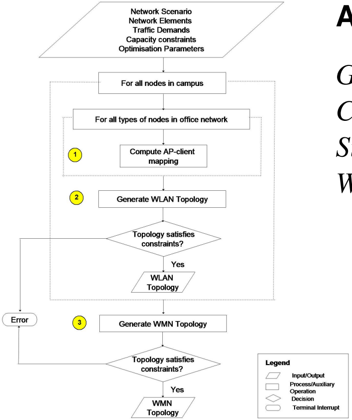
Three design stages





Problem formulation

Stage 1

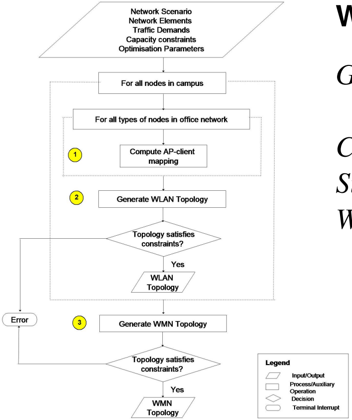


AP-assignment

Given client nodes
Compute APs required
Subject to capacity constraints
While minimizing |APs|

Problem formulation

Stage 2

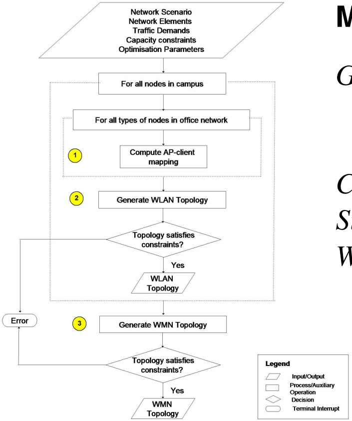


WLAN topology design

Given client nodes, deployment layout
Construct WLAN topology
Subject to capacity constraints
While minimizing network infrastructure (APs)

Problem formulation

Stage 3



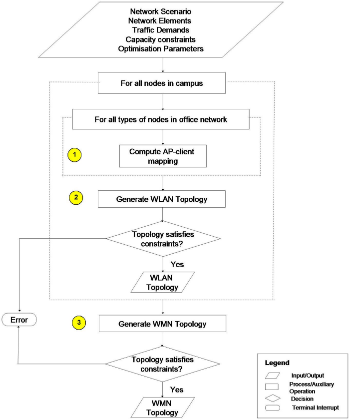
Mesh network design

Given deployment layout, AP nodes deployed and their characteristics
Construct backhaul topology
Subject to capacity constraints
While minimizing network infrastructure (mesh nodes and links)

Outline

Solution Approach

Stage 1

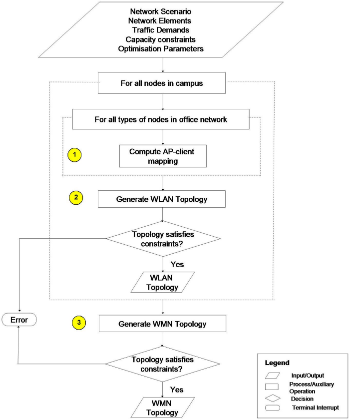


AP-assignment

- Analyse heterogeneous application deployments
- Prioritise applications to improve system utilisation
- Validate with simulation

Solution Approach

Stage 2

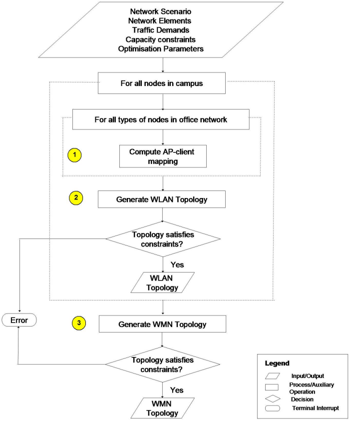


WLAN topology design

- Framework for deploying WLANs from simple network input parameters
- Construct topology using AP-assignment solutions as input
- Validate with simulation

Solution Approach

Stage 3

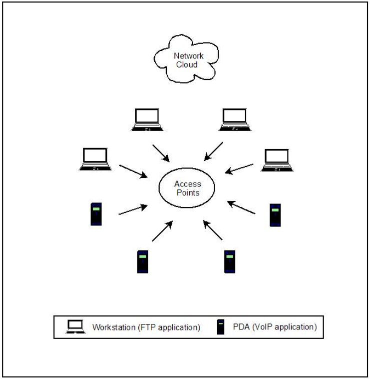


Mesh network design

- Framework for deploying WMNs from simple network input parameters
- Optimisation problem for Node locationing and topology construction
- Minimise network deployment cost using node and link costs

Outline

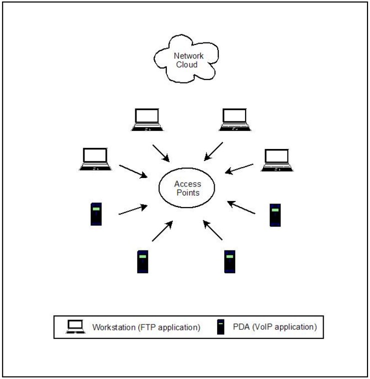
Associating Clients with APs



AP-assignment

Given client nodes
Compute APs required
Subject to capacity constraints
While minimizing |APs|

Capacity of WLANs



Aim

- Study single application scenario
  - Analyse 802.11 DCF mechanism
  - Realtime applications (voice and video codecs)
  - Theoretical vs Simulation results
- Capacity of system (in number of flows)
- Base case for analysis of heterogeneous deployments

System Setup

DCF schemes

Scheme	Data rate (in *Mbps*)
802.11b	1, 5.5, 11
802.11g	1, 11, 54

Codec parameters

↓ Parameters / Codecs →	G.711	G.723.1	G.729	GSM
Bit rate (in kbps)	64	6.4	8	13.2
Framing interval (in ms)	20	20	20	20
Payload (in bytes)	160	24	20	33

MAC parameters and Stack overheads

Theoretical Calculation
Terms

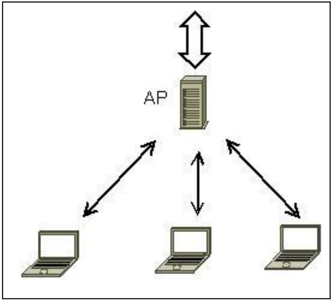
Term	Definition
pkt	Packet size (at MAC, in bytes)
ACK	Size of ACK packet (14 bytes for 802.11)
r	Data rate (in *Mbps*)
DIFS	DIFS time (in μS)
SIFS	SIFS time (in μS)
slot	Slot time (in μS)
backoff	Backoff
PHY	PHY overhead (in μS)

Throughput (T)

$$T = \frac{Payload}{t_{total}} = \frac{pkt * 8}{DIFS + SIFS + 2 * PHY + \frac{backoff}{2} * slot + t_{pkt} + t_{ack}}$$

Where,
$t_{pkt} = \frac{(pkt+MAC)*8}{r}$, $t_{ack} = \frac{ACK*8}{r}$

Simulation Setup



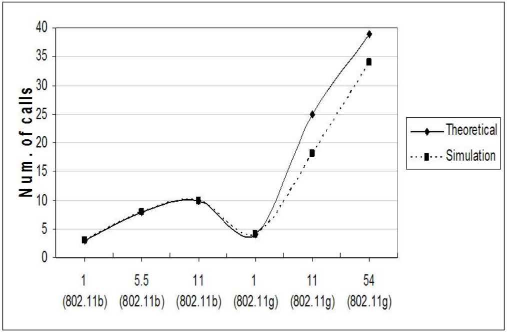
Implementation details

- Opnet Modeler
- Voice scenarios modeled as application definition
- Number of flows increased until constraints failed

Constraints

- Throughput satisfaction
- Delay $\leq$ 75 msec

Results: G.711 Codec
Theoretical vs Simulation



Scheme $\longrightarrow$ Data rate (r)	802.11b			802.11g		
	1	5.5	11	1	11	54
Theoretical	3	8	10	4	25	39
Simulation	3	8	10	4	18	34

Results: Voice Codecs

Scheme $\longrightarrow$ Data rate (r)	802.11b 1	5.5	11	802.11g 1	11	54
G.711	3	8	10	4	25	39
G.723.1	6	10	11	8	33	42
G.729	6	10	12	9	33	42
GSM	5	10	11	8	32	42

Table: Maximum number of voice calls: theoretical results.

Scheme $\longrightarrow$ Data rate (r)	802.11b 1	5.5	11	802.11g 1	11	54
G.711	3	8	10	4	18	34
G.723.1	7	11	11	7	23	36
G.729	6	11	11	7	22	36
GSM	6	10	11	7	22	35

Table: Maximum number of voice calls: simulation results.

Detailed calculation and graphs

Observations

- Simulation results closely follow theoretical results
- Theoretical results form upper bound
- 802.11g vs 802.11b: Effect of shorter timings seen in 11 Mbps case
- Delay $\ll$ Delay constraint (Max delay $\leq$ 18 μS)
- Minimal variation in number of calls between codecs
- CSMA/CA mechanism is main limitation
- Results well known [Anurag Kumar, Comm Networking, 2005]

Results: Video Capacity

Scheme $\longrightarrow$ Data rate (r)	802.11b 1	802.11b 5.5	802.11b 11	802.11g 1	802.11g 11	802.11g 54
SQCIF 128x96, 30fps	4	13	16	5	34	59
QCIF 176x144, 15fps	3	13	20	3	31	83
CIF 352x286, 10fps	1	6	10	1	13	46

Table: Maximum number of video flows: theoretical results.

Scheme $\longrightarrow$ Data rate (r)	802.11b 1	802.11b 5.5	802.11b 11	802.11g 1	802.11g 11	802.11g 54
SQCIF 128x96, 30fps	4	13	16	5	29	76
QCIF 176x144, 15fps	3	13	20	3	27	94
CIF 352x286, 10fps	1	6	10	1	14	52

Table: Maximum number of video flows: simulation results.

Detailed calculation and graphs

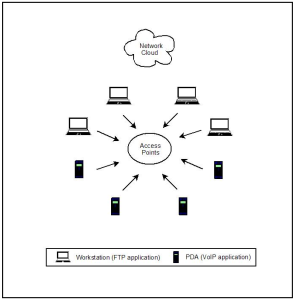
Comments

- Homogeneous applications can be provisioned in DCF
- Realtime applications can be provided QoS guarantees - voice and video
- AP bottleneck: Equal opportunity CSMA/CA leads to AP starvation
- Heterogeneous deployment difficult
  - Single FTP flow breaks delay constraint (G.711 max calls scenario) ▸ Extending DCF
- 802.11e standard for QoS provisioning
  - Complex standard, difficult to implement
  - Not widely adopted
  - Wireless MultiMedia (WMM) uses parts of 802.11e

Homogeneous analysis forms base case for analysis of heterogeneous deployments

Outline

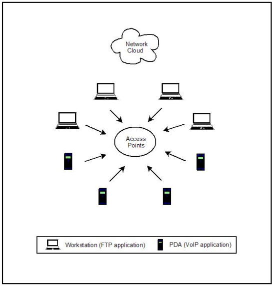
Problem Statement: Recap



AP-assignment

Given *heterogeneous* client nodes
Compute APs required
Subject to capacity constraints
While minimizing |APs|

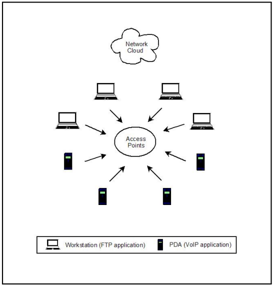
Deploying Heterogeneous Applications I



Issues with homogeneous capacity analysis

- Network utilisation is not maximal (On average, number of flows less than maximum flows)
- Homogeneous capacity unrelated to heterogeneous capacity
- WLAN capacity usually evaluated as maximum capacity

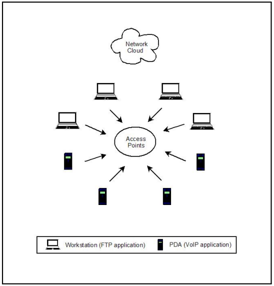
Deploying Heterogeneous Applications II



Heterogeneous capacity analysis

- Capacity in terms of heterogeneous applications
- Analysis of realtime applications with non-realtime applications
- Example: VoIP and FTP deployment

Deploying Heterogeneous Applications III

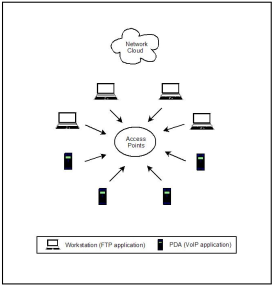


Sub-optimal heterogeneous application deployment

Deploy restricted number of priority applications

- Implement priority mechanism
- Number of flows = k ($<$ n, where n = homogeneous capacity)

Deploying Heterogeneous Applications IV



Sub-optimal heterogeneous application deployment

Deploy additional non-prioritised applications

- Best effort service
- Applications can be of same class as priority applications

Use restricted number of flows to set ACL policies

Deploying Heterogeneous Applications V

Example Sub-optimal G.711 Calls

- 802.11b 11 Mbps, G.711 codec
- Theoretical capacity

$$
\begin{aligned}
T &= \frac{Payload}{t_{total}} \\
&= \frac{pkt * 8}{DIFS + SIFS + 2 * PHY + \frac{backoff}{2} * slot + t_{pkt} + t_{ack}} \\
&= \frac{200 * 8}{DIFS + SIFS + 2 * PHY + \frac{31}{2} * slot + 170.18 + 10.18} \\
&= \frac{1600}{934.36} = 1.712\ Mbps
\end{aligned}
$$

- G.711 bandwidth $b = 0.16\ Kbps$
- Maximum theoretical calls = $\lfloor T/b \rfloor$ = 10 calls

Deploying Heterogeneous Applications VI

- Sub-optimal capacity

$$\lfloor k.T/b \rfloor = \lfloor 1.73k/.16 \rfloor$$

- Example: 30% bandwidth reservation for voice calls
  $\lfloor 1.73k/.16 \rfloor = \lfloor 1.73 * 0.3/.16 \rfloor = 3\ calls$

Sub-optimal Capacity: G.711 deployment

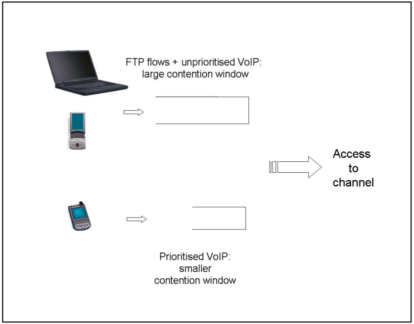
k ↓	Number of calls: $\lfloor kT/b \rfloor$					
	802.11b (in mbps)			802.11g (in mbps)		
	1	5.5	11	1	11	54
$\frac{T}{b} \rightarrow$	3.805	9.135	10.818	4.781	25.782	39.651
1.0	3	9	10	4	25	39
0.9	3	8	9	4	23	35
0.8	3	7	8	3	20	31
0.7	2	6	7	3	18	27
0.6	2	5	6	2	15	23
0.5	1	4	5	2	12	19
0.4	1	3	4	1	10	15
0.3	1	2	3	1	7	11
0.2	0	1	2	0	5	7
0.1	0	0	1	0	2	3

Table: k vs Number of voice calls for G.711 codec.

▸ Sub-optimal capacity calculations

Sub-Optimal Application Deployment

Problem definition

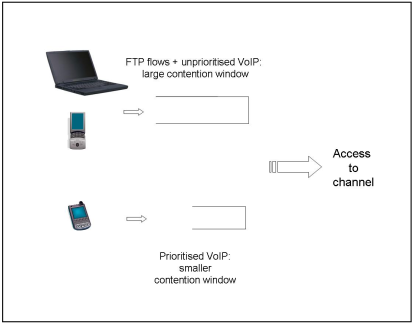


Application classes

- *Alpha* (α): Prioritised applications under ACL
- *Beta* (β): Applications with normal priority
- *Gamma* (γ): Applications of same class as *Alpha* running un-prioritised

Sub-Optimal Application Deployment

Problem definition



SOAP1

Given k Alpha flows $(|\alpha| = k)$
Compute number of *Beta* flows $(|\beta|)$
Subject to constraints ***R***

▸ SOAP2

Sub-Optimal Application Deployment

Implementation details

- Contention-window based service differentiation mechanism
- Impose ACL mechanism on α flows
- Add additional β andor γ flows as best effort service
- Extension of DCF MAC in OPNET Modeler
- Constraints ***R***:
  - α: Throughput and delay constraints
  - β, γ: Throughput constraint

Application	CWmin	CWmax
VoIP (priority)	15	31
FTP	31	1023

Table: Contention window parameters for SOAP.

Sub-Optimal Application Deployment
Simulation setup

- 802.11g mechanism
- G.711 codec
- Application classes

Application class	Application	
α	VoIP -	G.711
β	FTP -	250 & 500 Kbps
γ	VoIP -	G.711

- Constraints ***R***:
  - For all classes: Throughput satisfaction
  - α: $\alpha_k < 75ms$

Other simulation parameters

Sub-Optimal Application Deployment

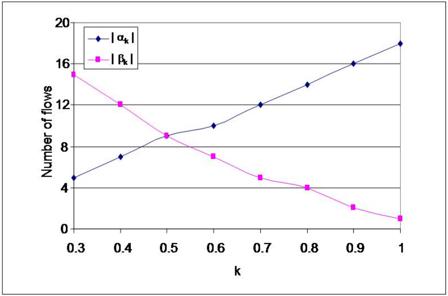
Results

k	$\|\alpha_k\|$	α_k delay (in s)	$\|\beta_k\|$	β_k throughput (in bps)	β_k delay (in s)
1.0	18	*0.086*	*1*	*101247*	*0.008*
0.9	16	0.070	2	758230	0.105
0.8	14	0.073	4	1481418	0.013
0.7	12	0.073	5	2229776	0.015
0.6	10	0.072	7	2969675	0.015
0.5	9	0.071	9	3386316	0.016
0.4	7	0.038	12	4293402	0.022
0.3	5	0.011	15	5179227	0.021

▸ Other results

Sub-Optimal Application Deployment

Observations



- $\alpha =$ G.711 voice codec and $\beta =$ FTP 500 Kbps
- At $k = 0.4$ effect of β on α negligible
- System utilisation improves from 30% to 50%
- Table used to set ACL - operating point of AP

Stage Summary

- Theoretical and simulation study of homogeneous and heterogeneous deployments
- Joint deployment of realtime and non-realtime applications
- Application prioritisation for sub-optimal application deployment
- System utilisation improvement $\sim$ 75% over normal DCF (with `SOAP1`)
- Access control limit mechanism for AP management

SOAP improves system utilisation

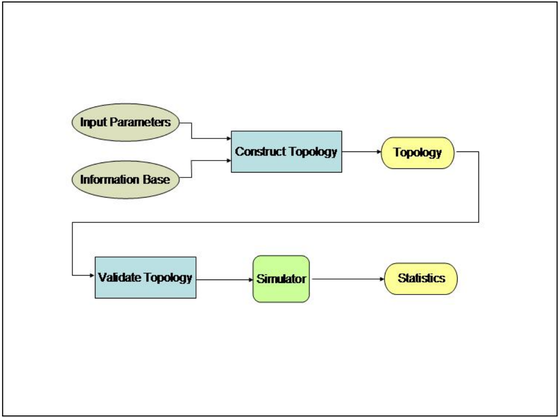
Problem Definition

WLAN topology design problem

Given client nodes & deployment area,
Construct WLAN topology,
Subject to capacity constraints,
While minimizing nw infrastructure (num of APs).

Outline

Generic Framework



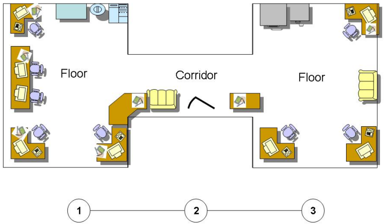
Advantages

- Allows planning for capacity at design stage
- Automate design process
- Eases validation with simulation

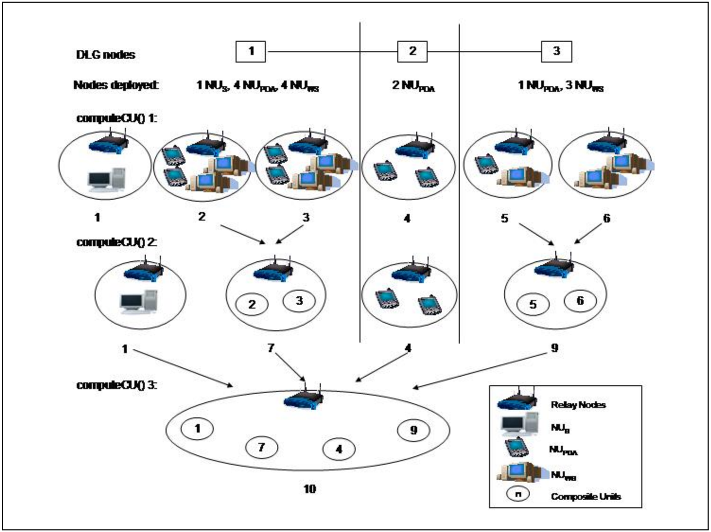
Outline

Example I

Office layout: (a) floor plan, (b) corresponding deployment layout



Example II
Topology construction



Outline

Composite Unit I

Definition

Virtual network element constructed for aggregating nodes, or branch of network, and their properties

$$CU = (CU' \mid NU)^{+}$$

Where,
CU = Composite Unit
NU = Node Unit (any network element)

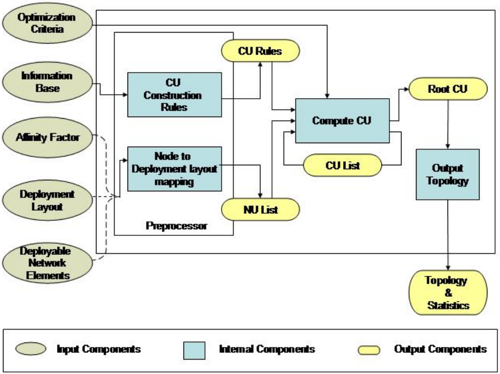
Composite Unit II
Class definition

class CU

private:	public:

int id;
string name;
 	
double outLoadTotal;
double inLoadTotal;
 	
LinkList* linkList;
ASList* asList;
CUList* childList;
	void print(int tab);
	CU(NodeType* nt);//NU constructor
	CU(); //CU constructor
	
	ASList* getASList();
	LinkType* getBestLink();
	void addChild(CU* cu);
	void rstChildProperty(LinkType*);
	void setProperty();
	void resetLinks(LinkType*);
	void resetTraffic();
	
	LinkList* getUnusedLinks();
	ASList* getUnfulfilledTraffic();

Wireless Infrastructure Network Deployment Tool (WIND)

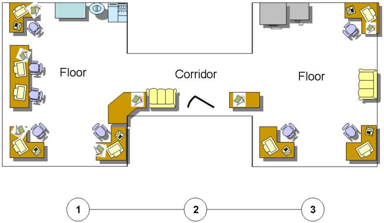


- Implemented using C++
- Input and output descriptions correspond with OPNET Modeler XML formats(for validation)

Pseudo code for WIND

Validation I

Deployment layout

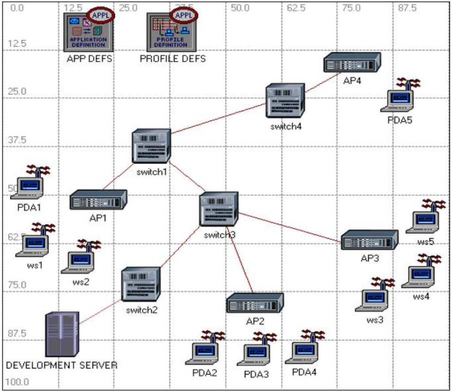


Example parameters

- 5 PDAs running a VoIP call (Load 100 Kbps)
- 5 Workstations running FTP client (Load 1000 Kbps)

▸ Information base and affinity factor

Validation II
Constructed topology



Simulation results

- Average VoIP throughput $\sim$ 100 Kbps
- Average FTP throughput $\sim$ 1000 Kbps

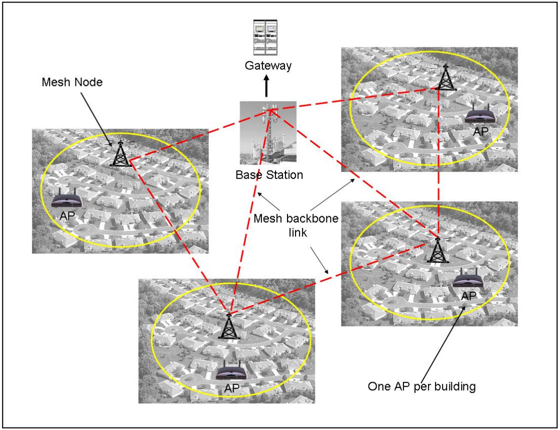
Stage summary

- Framework for deploying WLANs from simple network input parameters
- Inputs and Outputs modeled on simulator formats for integration
- Validation with simulation

Topology construction tool for WLANs

Outline

Example Campus Mesh Network



- Each building represents a WLAN
- APs connected to mesh with AP-mesh links
- Mesh nodes provide routes to gateway (through mesh links)
- AP-mesh forms a two-tier architecture

Mesh Network Design Problem

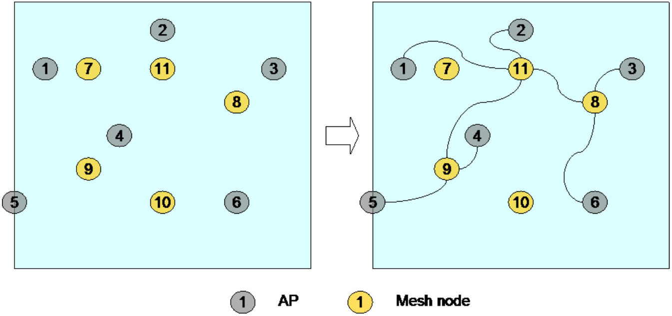
Mesh network design problem

Given deployment layout, AP nodes and their characteristics
Construct backhaul topology,
Subject to demand constraints
While minimizing network infrastructure (mesh nodes and links)

Constraints

- Capacity: Satisfy demand placed by APs (& their underlying networks)
- Cost: Minimise mesh nodes and links
- Connectivity: Connect all APs

Example Deployment: 6 APs, 5 Mesh Nodes

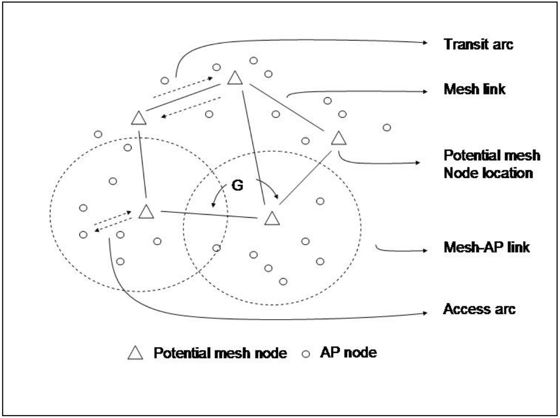


Deployment details

- Potential mesh nodes = 5
- Transmission range AP = 1.5 and mesh = 2
- Upper bound on mesh links (G) = 4
- Demands (100 Kbps) = $<1-2>, <2-5>, <2-6>, <3-4>, <3-6>, <4-6>$ & $<5-2>$

Outline

Network Model



Requirements

- Determine potential links (Mesh and Mesh-AP)
- Node and link costs
- Objective function
- Constraints

Computing Potential Links

- Distance-based: Compute distance between nodes and compare with transmit radius of AP
  Example:
  Given $AP = (x, y, r, \ldots), Mesh = (x', y', r', \ldots)$
  Potential link condition:

$$\sqrt{(x - x')^2 + (y - y')^2} < r$$

- Power-based: Compute distance between nodes using transmit power

▸ Calculating potential links using channel conditions

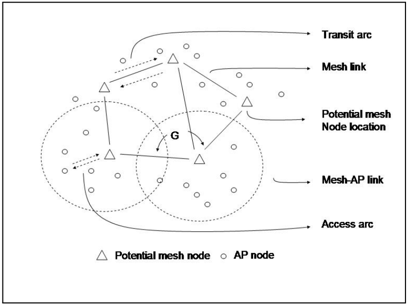
Node and Link Costs

- φ_v: cost of installing mesh node *v*.
- κ_e: cost of installing link *e*.

Cost of link

- Cost of hardware (σ_e)
- Cost of power requirements (determined by transmit power)
  - Fixed power: $\kappa_e = \sigma_e + ceil\ (\ r_e^2/\rho_e\ )$
    Where,
    r_e is transmit radius of node in link *e*
    ρ_e is a cost factor
  - Variable power: $\kappa_e = \sigma_e + ceil\ (\ tx\_dist_e^2/\rho_e\ )$
    Where,
    tx_dist_e is transmission distance
    ρ_e is a cost factor

Objective Function



Objective function

Minimize

$$F = \sum_{e} \kappa_e u_e + \sum_{v} \varphi_v s_v$$

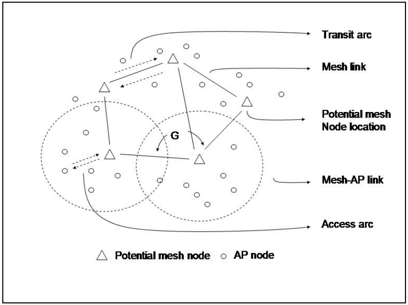
Where,

u_e = binary variable specifying whether link e is ON/OFF

s_v = binary variable specifying whether node v is ON/OFF

▸ Mesh topology design formulation

Constraints



Demand constraints

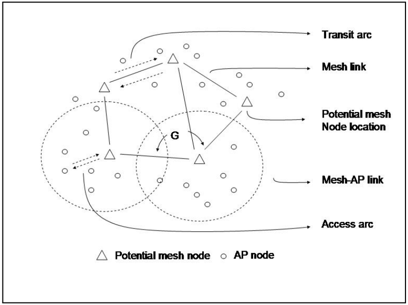
- Total demand flowing on each link not to exceed link capacity (1,5)
- Each demand has path from source AP to destination (2,3,4)
- Upper bound on number of demands per AP

Link constraint

- Upper bound on the number of links per node - *G* (6)

Mesh topology design formulation

Comments



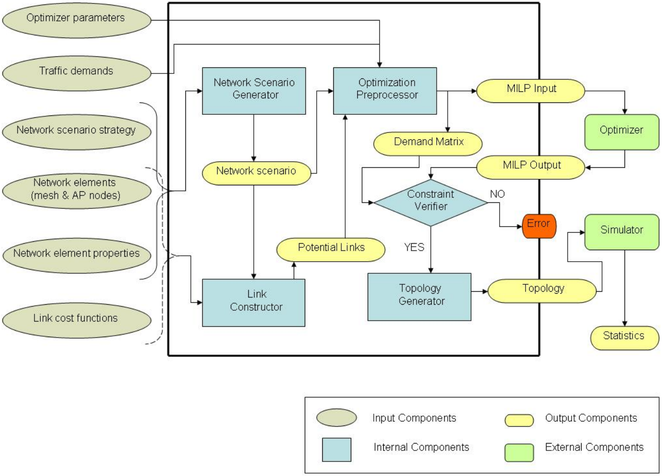
- Modeling of nodes and links as binary variables
- Mixed-Integer Linear Programmming problem (MILP)
- Finds node location and topology
- Routing algorithm computes all pairs shortest path

Routing algorithm

Outline

Extending WIND

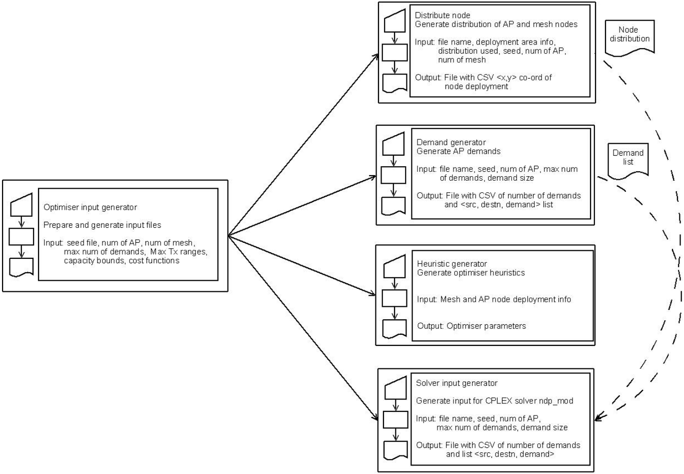
$\texttt{WIND}_{\texttt{wmn}}$ tool overview



▸ Input parameter details ▸ Module details

Implementation Details

- Implemented using PERL and ILOG OPL
- CPLEX solver used for MILP formulation



Experiment Details

Parameter	Value
Area	100mx100m
AP/Mesh Tx Range	70m
Mesh node cost φ	1000
Mesh link cost factor ρ	10
Max. Links G	4
Link capacity	10 Mbps
Demand	1 Mbps

- Mesh and AP nodes deployed randomly
- 11 artificially generated loads for each network scenario

Results

AP	Potential mesh	Exec time (s)	Mesh nodes (min,max)	Links (min,max,avg)
8	5	< 1	2, 3	8, 10, 10
10	7	50.93	3, 4	10, 13, 12
10	8	69.86	3, 4	10, 13, 12
12	7	178.12	3, 6	12, 16, 15
12	8	854.51	3, 5	12, 16, 15

- Average number of links: avg = ceil(average of all scenarios)

Stage Summary

- Framework for deploying WMNs from simple network input parameters
- Node locationing and topology construction
- Minimise network deployment cost using node and link costs

Node locationing and topology construction tool for WMNs

Summary

Contributions

- Provisioning 802.11 WLANs in homogeneous and heterogeneous scenarios.
- Capacity-constrained design of wireless networks.
- WIND tool for design of local area and backhaul wireless networks.

Possible extensions

- Include coverage as constraint in design problem
- Scheduling and routing issues in WMN design
- Use of tool in other areas: Sensor networks (lifetime constraint), Sparse networks (reachability)

Publications

- Automatic topology generation for a class of wireless networks. *IEEE International Conference On Personal Wireless Communications*, 2005. Joint work with: Sridhar Iyer.
- Automated design of VoIP-enabled 802.11g WLANs. *OPNETWORK*, 2005. Joint work with: Sridhar Iyer and Atanu Guchhait.
- Designing multi-tier wireless mesh networks: Capacity-constrained placement of mesh backbone nodes. *World Wireless Congress*, 2006. Joint work with: Sridhar Iyer.
- Capacity-constrained design of resilient multi-tier wireless mesh networks. *IEEE Infocom Student Workshop*, 2006. Joint work with: Sridhar Iyer.
- WIND: A Tool for capacity-constrained design of resilient multi-tier wireless mesh networks. *IEEE Infocom Poster Session*, 2006. Joint work with: Sridhar Iyer.
- Bridging the gap between reality and simulations: An Ethernet case study. *IEEE International Conference on Information Technology*, 2006. Joint work with: Punit Rathod and Srinath Perur.
- VoIP-based intra-village teleconnectivity: An architecture and case study. *First annual workshop on Wireless Systems: Advanced Research and Development (WISARD)*, 2006.
  Joint work with: Janak Chandarana, K. Sravana Kumar, Srinath Perur, Sameer Sahasrabuddhe and Sridhar Iyer.

MAC Parameters and Stack Overheads

802.11 DCF MAC parameters

Parameter (in μS)	802.11b	802.11g
Slot time	20	9
SIFS	10	10
DIFS (= SIFS + 2 * Slot time)	50	28
PHY preamble	192	20
Signal extension	-	6

Table: 802.11 b and g MAC parameters: timing, preamble transmission time and signal extension.

Stack overheads

Overhead	Value (in bytes)
RTP	12
UDP	8
IP	20
MAC	34

Table: RTP, UDP, IP and MAC stack overheads.

Voice Capacity: Maximum Calls I

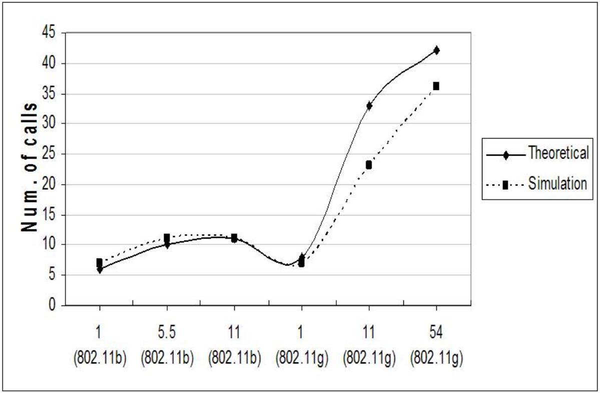


Figure: Maximum G.723.1 voice calls: theoretical vs simulation results.

Voice Capacity: Maximum Calls II

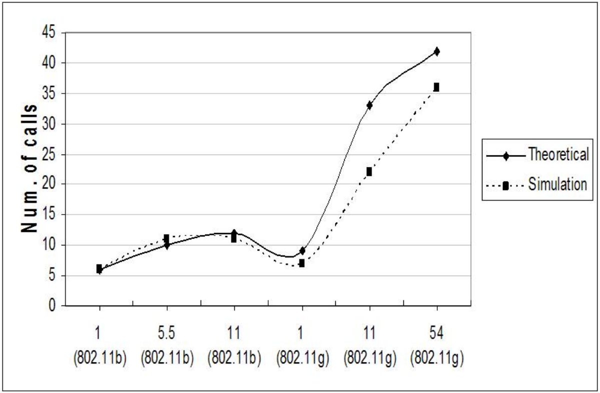


Figure: Maximum G.729 voice calls: theoretical vs simulation results.

Voice Capacity: Maximum Calls III

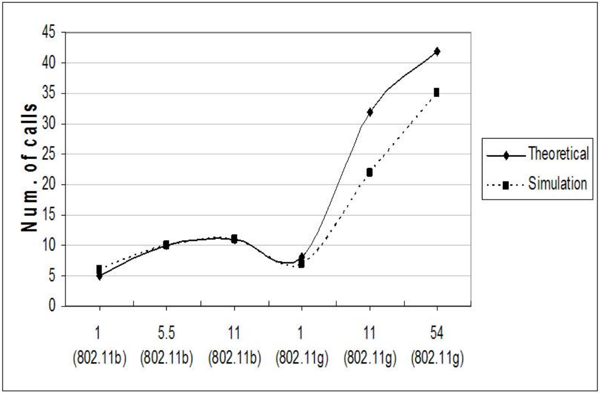


Figure: Maximum GSM voice calls: theoretical vs simulation results.

Voice Capacity: Maximum Calls IV

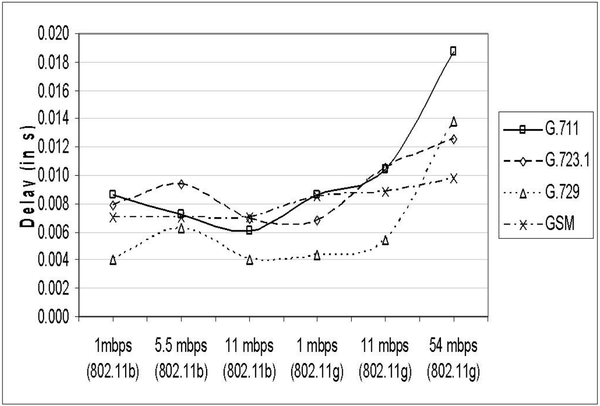


Figure: Delay for voice schemes in 802.11b/g.

Go back

Voice Capacity: Detailed Calculations

Scheme → Data rate (r)		802.11b 1	802.11b 5.5	802.11b 11	802.11g 1	802.11g 11	802.11g 54
pkt (in bytes)	G.711	200	200	200	200	200	200
	G.723.1	64	64	64	64	64	64
	G.729	60	60	60	60	60	60
	GSM	73	73	73	73	73	73
DIFS		50	50	50	28	28	28
SIFS		10	10	10	10	10	10
PHY		192	192	192	20	20	20
backoff		31	31	31	31	31	31
slot		20	20	20	9	9	9
t_{pkt} (in μs)	G.711	1872	340.364	170.182	1872	170.182	34.667
	G.723.1	784	142.546	71.273	784	71.273	14.519
	G.729	752	136.727	68.364	752	68.364	13.926
	GSM	856	155.636	77.818	856	77.818	15.852
t_{ack}	(in μs)	112	20.364	10.182	112	10.182	2.074
Throughput (T) (in Mbps)	G.711	0.584	1.435	1.712	0.727	4.022	6.293
	G.723.1	0.310	0.558	0.613	0.460	1.713	2.187
	G.729	0.297	0.527	0.577	0.444	1.621	2.056
	GSM	0.339	0.628	0.694	0.493	1.912	2.481
Bandwidth (b) (in Mbps)	G.711	0.160	0.160	0.160	0.160	0.160	0.160
	G.723.1	0.051	0.051	0.051	0.051	0.051	0.051
	G.729	0.048	0.048	0.048	0.048	0.048	0.048
	GSM	0.058	0.058	0.058	0.058	0.058	0.058
Number of calls	G.711	3	8	10	4	25	39
	G.723.1	6	10	11	8	33	42
	G.729	6	10	12	9	33	42
	GSM	5	10	11	8	32	42

Table: Number of voice calls: voice capacity calculations.

Go back

Voice Capacity: 39 Call Scenario

Simulation: 802.11g - G.711 codec

- Maximum 39 voice calls
- Packet drop $\leq$ 20 %
- Delay bounded

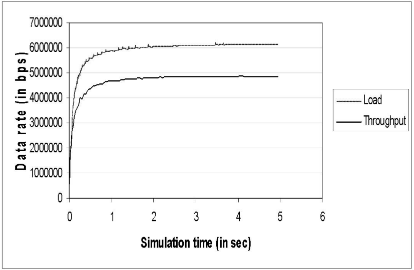


Figure: Load and throughput for G.711, 54 mbps 802.11g - 39 call scenario.

Go back

Video Capacity: Theoretical Calculation

Throughput equation: Extension for large payloads

- Maximum MAC payload size = 2304 bytes
- Large packets are fragmented
- Depending on codec, video packets may be fragmented

$$T_{frag} = \frac{Payload}{\frac{backoff}{2} * slot + t_{frag} * frag\_num}$$

Where,

$t_{frag} = DIFS + SIFS + 2 * PHY + t_{pkt_{frag}} + t_{ack}$

$t_{pkt_{frag}} = \frac{(pkt_{frag} + MAC) * 8}{r}$

$frag\_num = \lceil pkt / pkt_{frag} \rceil$

Go back

Video Capacity: Maximum Calls I

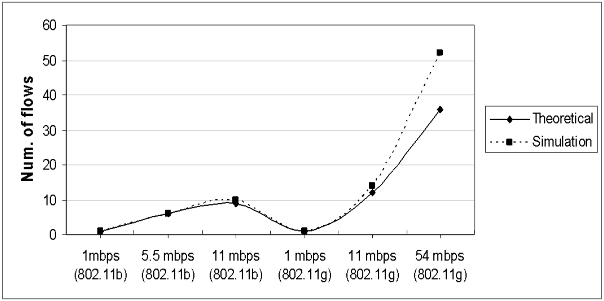


Figure: Maximum CIF video flows: theoretical vs simulation results.

Video Capacity: Maximum Calls II

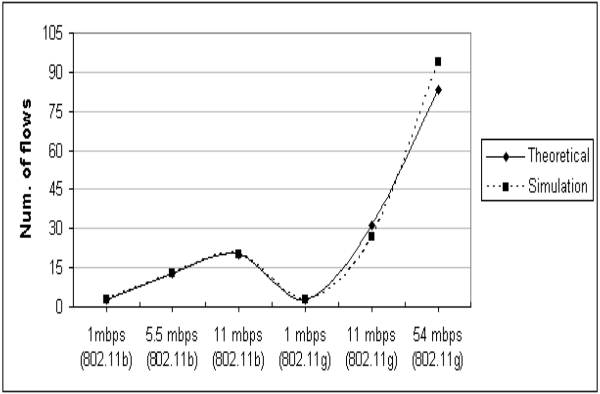


Figure: Maximum QCIF video flows: theoretical vs simulation results.

Video Capacity: Maximum Calls III

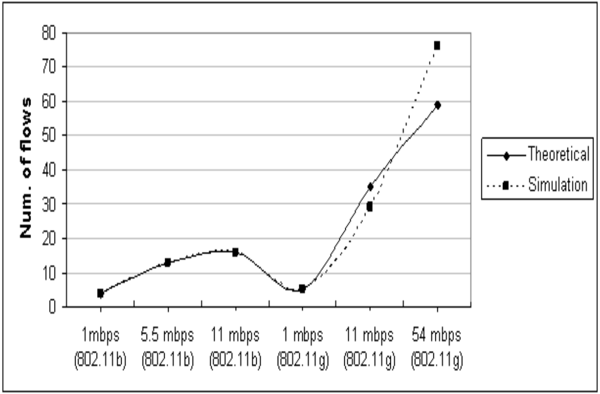


Figure: Maximum SQCIF video flows: theoretical vs simulation results.

Video Capacity: Maximum Calls IV

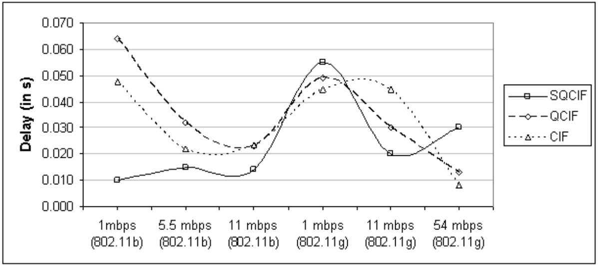


Figure: Delay for video schemes in 802.11b/g.

Go back

Video Capacity: Observations

- Large packet size affects maximum number of flows
- Maximum number of flows varies with codec (unlike Voice codecs)
- Efficient use of channel due to large packet size

Go back

Video Capacity: Detailed Calculations

Scheme → Data rate (r)		802.11b 1	802.11b 5.5	802.11b 11	802.11g 1	802.11g 11	802.11g 54
pkt	SQCIF	304	304	304	304	304	304
(in bytes)	QCIF	1112	1112	1112	1112	1112	1112
	CIF	3256	3256	3256	3256	3256	3256
frag_size		1500	1500	1500	1500	1500	1500
fragments	SQCIF	1	1	1	1	1	1
per pkt	QCIF	1	1	1	1	1	1
	CIF	3	3	3	3	3	3
DIFS		50	50	50	28	28	28
SIFS		10	10	10	10	10	10
PHY		192	192	192	20	20	20
backoff		31	31	31	31	31	31
slot		20	20	20	9	9	9
t_{pkt}	SQCIF	2704	491.636	245.818	2710	251.818	56.074
(in μs)	QCIF	9168	1666.909	833.455	9174	839.455	175.778
	CIF	12000	2181.818	1090.909	12006	1096.909	228.222
t_{ack}	(in μs)	1.978	0.36	0.18	7.978	6.18	6.037
Throughput	SQCIF	0.681	1.921	2.408	0.799	5.009	8.635
(T)	QCIF	0.887	3.644	5.568	0.936	8.290	22.165
(in Mbps)	CIF	0.686	3.281	5.267	0.709	7.016	24.065
Bandwidth (b)	SQCIF	0.146	0.146	0.146	0.146	0.146	0.146
(in Mbps)	QCIF	0.267	0.267	0.267	0.267	0.267	0.267
	CIF	0.521	0.521	0.521	0.521	0.521	0.521
Number of calls	SQCIF	4	13	16	5	34	59
	QCIF	3	13	20	3	31	83
	CIF	1	6	10	1	13	46

Table: Number of video flows: video capacity calculations.

Extending DCF to provide guarantees I

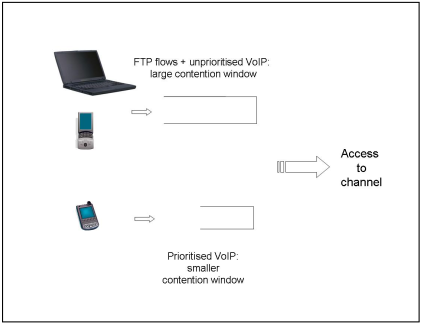


Figure: Contention window for ACL scheme - VoIP + FTP flows.

Extended DCF

- Simple scheme to differentiate traffic flows
- Prioritise realtime time applications
- Additional flows of best effort service

Extending DCF to provide guarantees II

Application	CWmin	CWmax
VoIP (priority)	15	31
FTP	31	1023

Table: Contention window parameters for SOAP.

Attribute	Value
Command mix (get/total)	50%
Inter-request time (s)	exponential(60)
File size (bytes)	constant(125000)
Fragmentation size (bytes)	1500
Type of service	Best Effort (AC_BE)

Table: FTP simulation parameters.

Extending DCF to provide guarantees III

Scheme	Load (in bps)	Throughput (in bps)	Delay (in sec)
DCF	81265	73712	0.009
Extended DCF	89575	85612	0.019

Table: Comparison of VoIP plus FTP performance DCF.

Observations

- FTP flows in extended DCF = 4
- VoIP delay in extended DCF $\leq$ 0.062s
- VoIP delay in DCF $\geq$ 0.1s

Go back

Sub-optimal Capacity: Voice Codecs

k ↓	Number of calls: $\lfloor kT/b \rfloor$ 802.11b (in mbps) 1	5.5	11	802.11g (in mbps) 1	11	54
$\frac{T}{b} \rightarrow$	6.494	11.15	12.115	9.966	34.608	43.093
1.0	6	11	12	9	34	43
0.9	5	10	10	8	31	38
0.8	5	8	9	7	27	34
0.7	4	7	8	6	24	30
0.6	3	6	7	5	20	25
0.5	3	5	6	4	17	21
0.4	2	4	4	3	13	17
0.3	1	3	3	2	10	12
0.2	1	2	2	1	6	8
0.1	0	1	1	0	3	4

Table: k vs Number of voice calls for G.723.1 codec.

Sub-optimal Capacity: Voice Codecs

k ↓	Number of calls: $\lfloor kT/b \rfloor$ 802.11b (in mbps) 1	5.5	11	802.11g (in mbps) 1	11	54
$\frac{T}{b} \rightarrow$	6.631	11.222	12.157	10.294	34.96	43.204
1.0	6	11	12	9	34	43
0.9	5	10	10	9	31	38
0.8	5	8	9	8	27	34
0.7	4	7	8	7	24	30
0.6	3	6	7	6	20	25
0.5	3	5	6	5	17	21
0.4	2	4	4	4	13	17
0.3	1	3	3	3	10	12
0.2	1	2	2	2	6	8
0.1	0	1	1	1	3	4

Table: *k* vs Number of voice calls for G.729 codec.

Sub-optimal Capacity: Voice Codecs

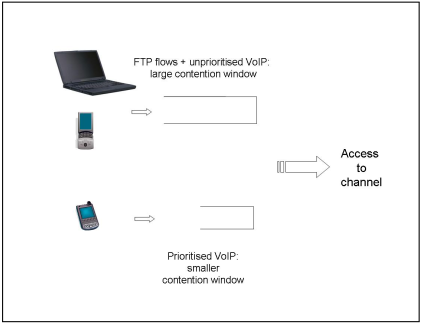
k ↓	Number of calls: $\lfloor kT/b \rfloor$ 802.11b (in mbps) 1	5.5	11	802.11g (in mbps) 1	11	54
$\frac{T}{b} \rightarrow$	6.204	10.989	12.019	9.298	33.841	42.847
1.0	6	10	12	9	33	42
0.9	5	9	10	8	30	38
0.8	4	8	9	7	27	34
0.7	4	7	8	6	23	29
0.6	3	6	7	5	20	25
0.5	3	5	6	4	16	21
0.4	2	4	4	3	13	17
0.3	1	3	3	2	10	12
0.2	1	2	2	1	6	8
0.1	0	1	1	0	3	4

Table: *k* vs Number of voice calls for GSM codec.

Go back

Sub-Optimal Application Deployment

Problem definition



SOAP2

Given k Alpha flows $(|\alpha| = k)$
Compute number of *Beta* flows $(|\beta|)$ *& Gamma* flows $(|\gamma|)$
Subject to constraints ***R***

▸ SOAP2 results ▸ Go back

Other Simulation Parameters

Attribute	Value
Command mix (get/total)	100%
Inter-request time (s)	exp(1)
File size (bytes)	FTP 250 - cons(31250)
	FTP 500 - cons(62500)
Fragmentation size (bytes)	1500
Type of service	Best Effort (AC_BE)

Table: SOAP simulation parameters for FTP - Average load 250 and 500 kbps.

Go back

SOAP1 Results I

k	$\lvert\alpha_k\rvert$	α_k delay (in s)	$\lvert\beta_k\rvert$	β_k throughput (in bps)	β_k delay (in s)
1.0	18	*0.086*	*1*	*63672*	*0.007*
0.9	16	0.067	3	713658	0.010
0.8	14	0.074	7	1447647	0.012
0.7	12	0.075	10	2181612	0.013
0.6	10	0.071	13	2920485	0.015
0.5	9	0.071	16	3306418	0.017
0.4	7	0.027	20	4134720	0.023
0.3	5	0.009	24	5002889	0.021

Table: SOAP1 results for FTP 250 Kbps with G.711 codec on 11 Mbps 802.11g.

SOAP1 Results II

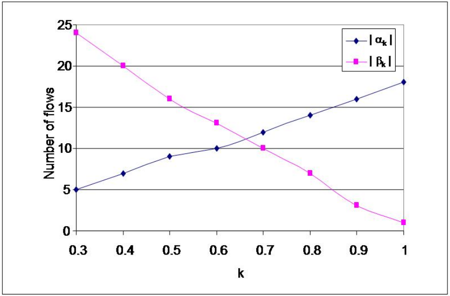


Figure: SOAP1 FTP 250 Kbps: k vs Number of α_k and β_k flows.

Go back

SOAP2 Results I

$\lvert\beta_k\rvert$	$\lvert\gamma_k\rvert$	γ_k delay	α_k delay
7 to 3	0	-	-
2	2	0.033	0.074
1	4	0.337	0.012

Table: SOAP2 results for $k = 0.8$ ($\lvert\alpha_{0.8}\rvert = 14$), FTP 250 Kbps with G.711 codec on 11 Mbps 802.11g. α_k and γ_k delays are in seconds.

Go back

$\mathtt{WIND}_{\mathtt{wlan}}$ Algorithm

input : ib: info base, ip: input parameters
$cuList \leftarrow NULL$
`// ` G_{DL}`:Deployment layout`
forall $v \in V(ip.G_{DL})$ **do**
    `// af:affinity factor`
    $deployedList \leftarrow \bigcup_{\forall i}(v.af_i * ip.num_{NU_i}).NU_i$
end
$cuList \leftarrow computeCU(cuList, ib)$
$printTopology(cuList)$

Algorithm 1: Pseudo-code for $\mathtt{WIND}_{\mathtt{wlan}}$

Algorithm: ComputeCU()

```
input  : cuList,ib: info base
if sizeOf(cuList) = 1) then return cuList
newCUList ← NULL
L ← linktypes_present(cuList)
forall lt ∈ L do
    cuList_lt ← cuList_lt + {cuList[i], cuList[i].linktype = lt}
    while cuList_lt NOTEMPTY do
        cuList' = lt.maxNodes(cuList_lt)
        // Average load
        t ← (Σ_∀j cuList'[j].total_load) / sizeOf(cuList')
        new cu'
        cu'.child(cuList')
        cuList_lt.remove(cuList')
        newcu_relay = findRelayNode(lt, t)
        cu'.child(cu_relay)
        for cu ∈ cuList' do cu.resetProperty()
        cu_relay.resetProperty()
        cu'.setProperty()
        newCUList.add(cu')
    end
end
return computeCU(newCUList, ib)
```

Algorithm 2: Psuedo-code for computeCU()

Go back

Information Base and Affinity Factor

NU	$Traffic_{out}$	$Traffic_{in}$	$Addr_{src}$	$Addr_{destn}$	Link	AS-Link Map	Mobility
NU_{PDA}	10000	100000	$< N_{PDA} >$	$< N_S >$	1	1	Yes
NU_{WS}	10000	1000000	$< N_{WS} >$	$< N_S >$	1	1	No
NU_S	10^6	100000	$< N_S >$	Undefined	2	2	No
NU_{Relay}	$5 * 10^5$	$5 * 10^5$	Undefined	Undefined	1,2	2	No
...	...	...	...	...	...	...	...

Table: Example information base. $Traffic_{out}$, $Traffic_{in}$ are in bits per second. Link type 1 represents a 802.11 10 Mbps wireless link and type 2 represents a 10 Mbps Ethernet link.

Node type ↓, Vertex →	**1**	**2**	**3**
NU_{PDA}	0.6	0.2	0.2
NU_{WS}	0.6	0	0.4
NU_S	1	0	0
...	...	...	...

Table: Affinity factors.

Potential Links: Including Channel Conditions

$$(\Psi_{rcv})_{dB} = (\Psi_{xmt})_{dB} - 10\eta \log_{10}(d/d_0) - \xi$$

Where,
$(\Psi_{rcv})_{dB}$ & $(\Psi_{xmt})_{dB}$ are received and transmit powers
d is transmit distance
d_0 is reference distance
η is path loss exponent
ξ is shadowing component

Go back

Mesh Topology Design Formulation

Variables:

x_{fw}: flow realising all demands originating at AP w on access arc f

x_{tw}: flow realising all demands originating at AP w on transit arc t

y_e: capacity of link e

$u_e = 1$ if link e is provided; 0, otherwise

$s_v = 1$ if mesh node v is installed; 0, otherwise

Objective function:

minimize $\boldsymbol{F} = \sum_e \kappa_e u_e + \sum_v \varphi_v s_v$

Constraints:

$\sum_t w_{et} \sum_w x_{fw} + \sum_f w_{ef} \sum_w x_{fw} \leq y_e$, $e = 1, 2, \ldots, E$ **- *(1)***

$\sum_f \beta_{fw} x_{fw} = H_w,\ w = 1, 2, \ldots, W$ **- *(2)***

$\sum_f \beta_{fw'} x_{fw} = -h_{ww'}$ **- *(3)***

$\sum_t \beta_{tv} x_{tw} + \sum_f \beta_{fv} x_{fw} = 0$ **- *(4)***

$y_e \leq M_e u_e$ **- *(5)***

$\sum_e \beta_{ev} u_e \leq G_v s_v$ **- *(6)***

Go back
Complete problem formulation

Mesh Topology Design Formulation

Indices:

$w = 1, 2, \ldots, W$: APs
$v = 1, 2, \ldots, V$: mesh nodes
$e = 1, 2, \ldots, E$: links
$f = 1, 2, \ldots, F$: directed access arcs (between AP & mesh nodes)
$t = 1, 2, \ldots, T$: directed transit arcs (between mesh nodes)

Mesh Topology Design Formulation

Constants:

$h_{ww'}$: volume of demand from AP w to w'

$H_w = \sum_{w'} h_{ww'}$: total demand outgoing from AP w

$\beta_{ev} = 1$ if link e is incident with mesh node v; 0, otherwise

$\beta_{fw} = -1$ if access arc f is incoming to AP w
$= 1$ if access arc f is outgoing from AP w
$= 0$ otherwise

$\beta_{fv} = -1$ if access arc f is incoming to mesh node v
$= 1$ if access arc f is outgoing from mesh node v
$= 0$ otherwise

$\beta_{tv} = -1$ if transit arc t is incoming to mesh node v
$= 1$ if transit arc t is outgoing from mesh node v
$= 0$ otherwise

$w_{ef} = 1$ if access arc f is realised on link e; 0, otherwise

$w_{et} = 1$ if transit arc t is realised on link e; 0, otherwise

κ_e : cost of installing link e

M_e : upper bound on the capacity of link e

φ_v : cost of installing mesh node v

G_v : upper bound on the number of radios of mesh node v

Mesh Topology Design Formulation

Variables:

x_{fw}: flow realising all demands originating at AP w on access arc f

x_{tw}: flow realising all demands originating at AP w on transit arc t

y_e: capacity of link e

$u_e = 1$ if link e is provided; 0, otherwise

$s_v = 1$ if mesh node v is installed; 0, otherwise

Objective function:

minimize $\boldsymbol{F} = \sum_e \kappa_e u_e + \sum_v \varphi_v s_v$

Constraints:

$\sum_t w_{et} \sum_w x_{fw} + \sum_f w_{ef} \sum_w x_{fw} \leq y_e$, $e = 1, 2, \ldots, E$ ***- (1)***

$\sum_f \beta_{fw} x_{fw} = H_w$, $w = 1, 2, \ldots, W$ ***- (2)***

$\sum_f \beta_{fw'} x_{fw} = -h_{ww'}$ ***- (3)***

$\sum_t \beta_{tv} x_{tw} + \sum_f \beta_{fv} x_{fw} = 0$ ***- (4)***

$y_e \leq M_e u_e$ ***- (5)***

$\sum_e \beta_{ev} u_e \leq G_v s_v$ ***- (6)***

Go back

Mesh Algorithm Formulation

cost_min ← *COSTMIN*
forall *ON/OFF combination of mesh_nodes* **do**
    `// on mesh nodes which have been switched ON`
    forall *ON/OFF combination of links & num_of_mesh_links* < *max_links* **do**
        forall *demands* **do**
            if *demand* < *remaining_link_capacity()* **then**
            cost ← *cost_of_shortest_path*() **if** *cost* < *cost_min* **then** *cost_min* ← *cost*
            adjust_link_capacity()
        end
    end
end

Algorithm 3: Psuedo-code for mesh routing.

Go back

WIND$_{wmn}$ Input Parameters

1. Network elements: Number of AP and potential mesh nodes
2. Network element properties: Properties of nodes and their associated links
3. Network scenario strategy: Properties of deployment layout and node distribution
4. Traffic demands: User generated traffic demands for each AP
5. Link cost functions: Cost functions for fixed and variable transmit powers
6. Optimizer parameters and heuristics: Heuristics and initial settings for the optimiser

Go back

WIND$_{wmn}$ Modules I

- Network scenario generator: Created based on deployment layout parameters and number of nodes. Creates locations of AP nodes and potential mesh nodes
- Link constructor: Uses heuristics to generate list of potential links
- Optimization preprocessor: Constructs inputs for optimiser and demand matrix for the constraint verifier

$\texttt{WIND}_{\texttt{wmn}}$ Modules II

- Optimizer: External optimizer invoked to solve MILP problem
- Constraint verifier: Verifies capacity constraints imposed on scenario by comparing optimizer output with demand matrix
- Topology generator: Constructs corresponding capacity-constrained topology
- Simulator: External simulator invoked to validate topology generated

Go back