Asymmetric Access Point for Solving the Unfairness Problem in WLANs

Andrzej Duda
with E. Lopez, M. Heusse, Y. Grunenberger, F. Rousseau, J. Casademont

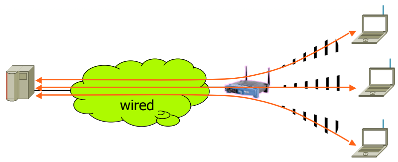
LIG
Grenoble Computer Laboratory

Iniquité dans les réseaux locaux sans fil : rétablir l'ordre juste

Outline

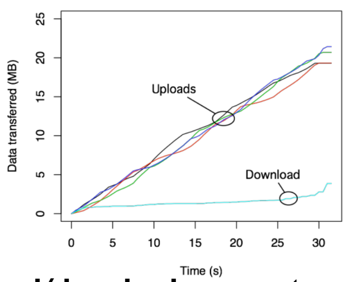
- Unfairness problem
- Idea of Asymmetric Access Point
- Idle Sense principles and properties
- Principles and performance of Asymmetric Access Point
- Conclusions

Unfairness problem



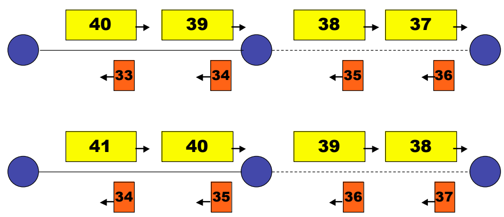
- TCP connections to mobile stations
  - download, upload
- Sporadic UDP traffic with real-time requirements (VoIP)
- We assume that wireless LAN is the bottleneck

Unfairness problem at TCP



- **Upload/download asymmetry**:
  - stations performing uploads obtain higher TCP throughput

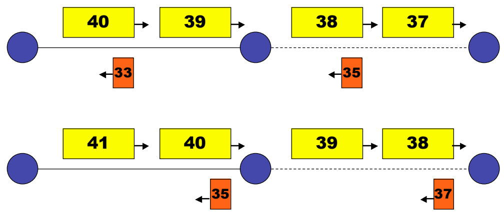
TCP Cumulative Acknowledgements



- A new cumulative ACK is generated only on receipt of a **new in-sequence** packet

i DATA i ACK

TCP Delayed Acknowledgements



- An ACK is **delayed** until
  - *k*-1 segments are received (*k*=2 typical)
  - *k* - nb. of data segments per ACK

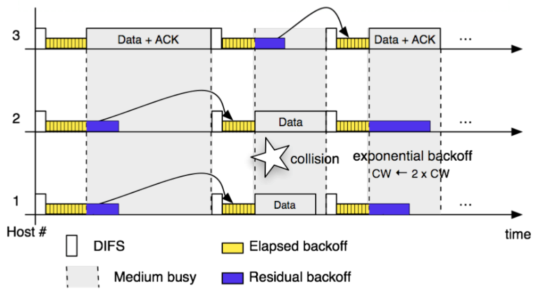
7

DCF characteristics

- Half-duplex operation
  - one frame at a time
- Equal channel access opportunities for all contending entities
  - AP and any of *N* stations
    - statistical share of 1/(*N*+1)
  - independent of frame length
- Exponential backoff
  - short term unfairness for larger *N*
  - increased unfairness if bad channel conditions

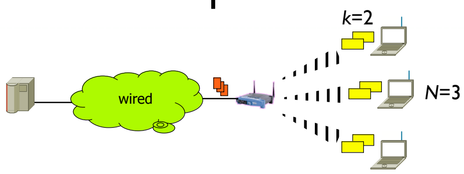
8

802.11 DCF in a nutshell



9

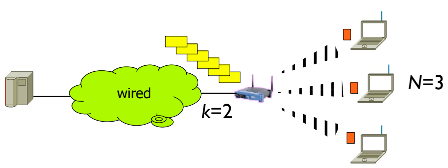
N uploads



- *kN* data segments at stations, *N* ACKs at AP
- AP share needs to be *N*/(*N* + *kN*), 1/3 for *k*=2
- If share of 1/(1+*N*)
  - short buffer at AP: loss, but ACKs are cumulative
  - large buffer at AP: longer RTT, limited by flow control

10

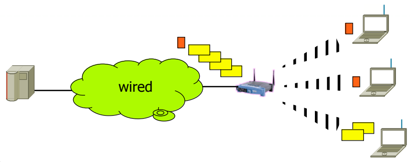
N downloads



- *kN* data segments at AP, *N* ACKs at stations
- AP share needs to be *kN*/(*N* + *kN*), 2/3 for *k*=2
- If share of 1/(1+*N*)
  - short buffer at AP: loss, limited by congestion control
  - large buffer at AP: longer RTT, limited by flow control

11

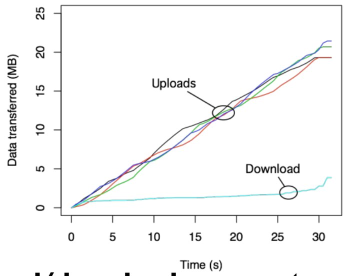
Mixed upload-download



- Common asymmetric situation
  - more downloads than uploads
- AP share needs to be something between 1/3 and 2/3 for *k*=2
- If 2/3, throughput limited anyway at sending stations

12

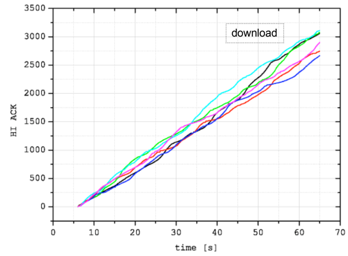
Unfairness problem at TCP



- **Upload/download asymmetry**:
  - stations performing uploads obtain higher TCP throughput

13

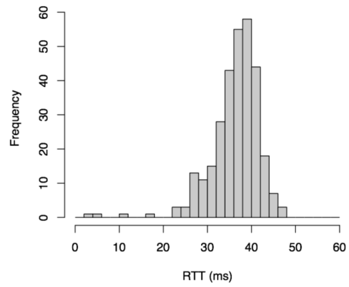
Unfairness problem at TCP



- **No unfairness**, but
  - **download bottleneck**: download flows saturate the AP buffer

14

Unfairness problem at TCP



- **Download bottleneck**: ping, 4 downloads + packet losses

15

Idea of AAP

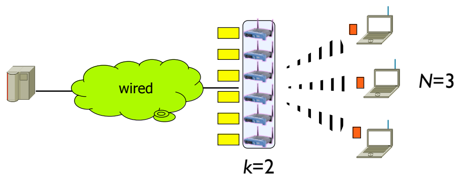
16

Approach

- Give more channel access opportunity to AP
  - *Asymmetric Access Point* benefits from *k* more share than all the stations in a cell (or *kN* than a single station, *k* - nb. of data segments per ACK)
  - corresponds to the worst case (*N* downloads)
  - increases performance in mixed upload/download scenario (unused capacity of AP is taken by stations)
  - to keep the buffer of AP empty so that TCP connections become self-clocked by the destination (short RTT over the wireless part)

17

Approach



- AAP behaves like *kN* contending entities
  - like means approximation
- AAP has *kN* more channel access opportunity than a single station

18

Solution

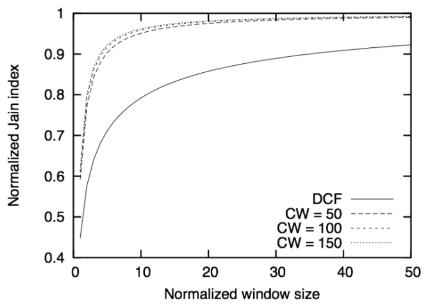
- *Asymmetric Access Point*
  - constant CW value, independent of *N*!
  - derived for given *k* and 802.11 variant
- Stations
  - operate according to *Idle Sense*
  - adapt CW to load conditions in the cell

Principles of Idle Sense

Towards Idle Sense

- Keep good aspects of DCF
  - No explicit information exchange
  - Keep backoff procedure: random backoff
- Modifications
  - No exponential backoff
    - make hosts use similar values of CW ⇒ fairness
  - Adapt CW to varying traffic conditions
    - more hosts, bigger CW; less hosts smaller CW
    - do not change CW upon frame loss

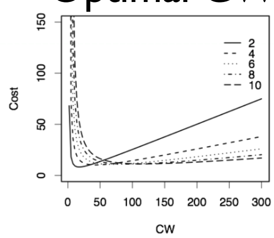
Why fixed Contention Window?



Idle Sense

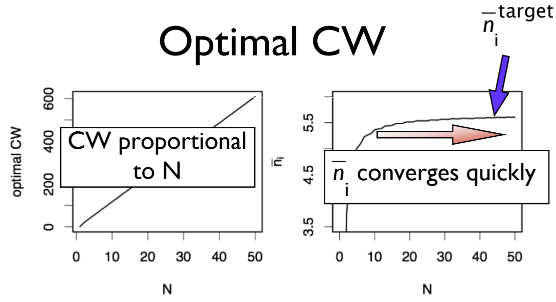
- Observe the number of idle slots
  - Channel load indicator
- Control CW
  - Adjust CW to the current state
  - Optimal operation in all conditions
    - What is the optimal CW?
    - How it relates to the number of idle slots?

Optimal CW



Cost function: Proportion of time spent in collisions or contention
Minimizing the cost ⇒ Maximizing throughput

Optimal CW

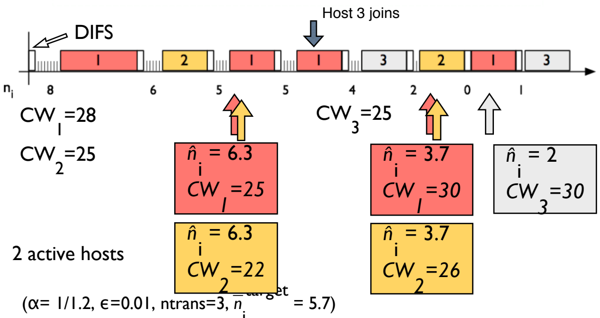


$\overline{n}_i$: average number of idle slots between transmission attempts

Idle Sense

- Hosts track $\overline{n}_i$ and make it converge to the target value
  - Each host estimates $\hat{n}_i$
  - Rises/Lowers CW when $\overline{n}_i$ too small/big compared to $\overline{n}_i^{target}$
  - Adjusting CW is done according to AIMD
  - all hosts converge to a similar value of CW

Example



(α= 1/1.2, ϵ=0.01, ntrans=3, n_i^{target} = 5.7)

Properties

- Contention control independent of frame loss detection
  - No "bad day" effect
  - Solves the physical layer capture effect
- Short term fair
- Fixes performance anomaly
  - Time fairness achieved by scaling CW according to the transmission rate
- No hardware modification required

Properties: Channel adaptation

- With Idle Sense, the collision probability P_c is known and bounded (after convergence)
  - Frame loss probability $P_{err} \approx 1 - P_c - P_{ok}$
    - P_{ok} can be observed
- Provides a new means for setting the right transmission rate
  - Change rate when P_{err} exceeds a given threshold
  - May be combined with SNR measurements

Principles and performance of AAP

CW of AAP

- Throughput
  - $X_{AAP} = kN \cdot X_{STA}$
- Probability of transmission
  - $Pt_{AAP} = kN \cdot Pt_{STA}$
- What we can control: Probability of transmission attempt
  - $Pe_{AAP} \approx kz/(kz+1)$, where
  - z is a solution of a non-linear equation
  - $CW_{AAP} = 2/Pe_{AAP} + 1$

Simulation, no AIMD

- 802.11b, *k*=2

N	CW_{AP}	CW_{STA}	X_{AP}	X_{STA}	$\sum X_{\mathrm{STA}}$	X	*AP share*
2	18.86	74.0	5.19	1.27	2.54	7.72	0.6714
4	18.86	148.0	5.19	0.63	2.52	7.71	0.6730
10	18.86	369.9	5.19	0.25	2.51	7.70	0.6738
15	18.86	554.9	5.19	0.17	2.51	7.69	0.6740
20	18.86	739.8	5.19	0.13	2.51	7.69	0.6741
25	18.86	924.8	5.19	0.10	2.51	7.69	0.6742

- 802.11g, *k*=2

N	CW_{AP}	CW_{STA}	X_{AP}	X_{STA}	$\sum X_{\mathrm{STA}}$	X	*AP share*
2	13.54	52.7	21.31	5.17	10.34	31.65	0.6734
4	13.54	105.4	21.32	2.56	10.24	31.56	0.6755
10	13.54	263.5	21.32	1.02	10.19	31.51	0.6767
15	13.54	395.2	21.32	0.68	10.17	31.49	0.6770
20	13.54	526.9	21.32	0.51	10.17	31.49	0.6771
25	13.54	658.7	21.32	0.41	10.16	31.49	0.6772

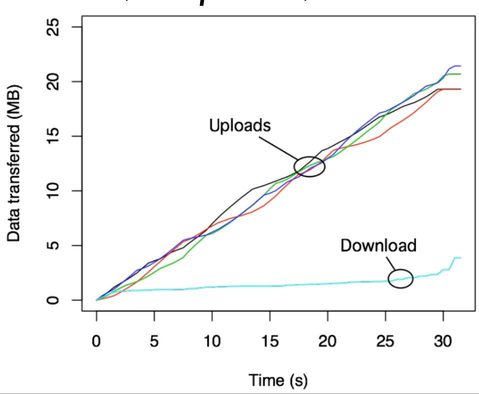
Simulation 802.11g, with AIMD

AAP, $k = 1$, $CW_{AAP} = 18$					
N	X_{AP}	X_{STA}	$\sum X_{\mathrm{STA}}$	X	*AP share*
2	18.48	6.84	13.67	32.15	0.5748
4	16.68	3.81	15.23	31.91	0.5226
10	15.20	1.64	16.41	31.61	0.4809
15	14.92	1.11	16.59	31.51	0.4735
20	14.73	0.84	16.74	31.47	0.4681
25	14.69	0.67	16.76	31.45	0.4672
AAP, $k = 2$, $CW_{AAP} = 13$					
N	X_{AP}	X_{STA}	$\sum X_{\mathrm{STA}}$	X	*AP share*
2	23.77	4.53	9.05	32.82	0.7242
4	22.34	2.59	10.35	32.69	0.6834
10	21.24	1.13	11.28	32.52	0.6531
15	21.19	0.75	11.32	32.50	0.6518
20	21.02	0.57	11.42	32.44	0.6481
25	20.98	0.46	11.46	32.44	0.6468

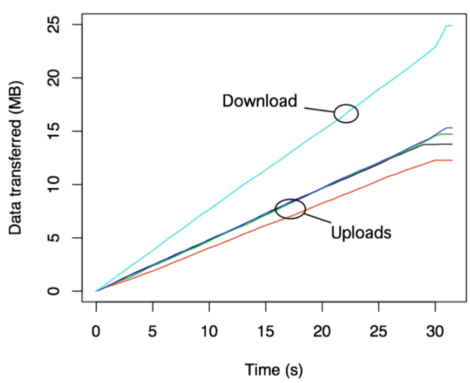
Measurements

- Implementation of Idle Sense
  - Intel PRO/Wireless 2200BG 802.11 a/b/g cards
  - Modified firmware, operational cards
- AP - FreeBSD box
  - constant CW
- Stations close to AP, good channel conditions, 802.11a at 54 Mb/s

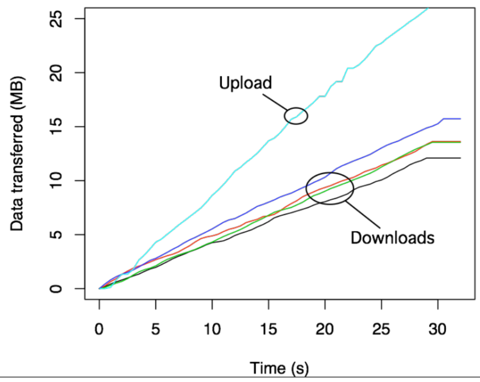
DCF, 4 *uploads*, 1 *download*



AAP, 4 *uploads*, 1 *download*

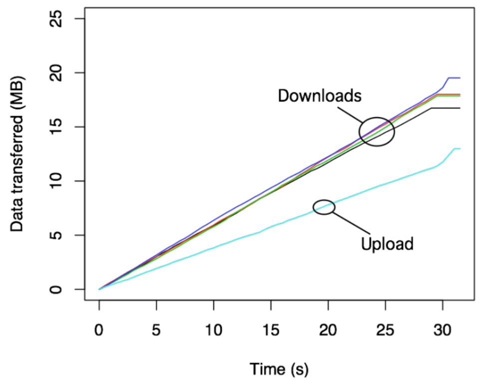


DCF, 1 *upload*, 4 *downloads*



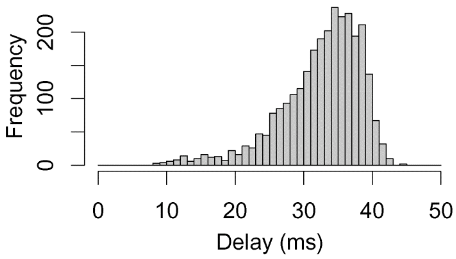
37

AAP, 1 *upload*, 4 *downloads*



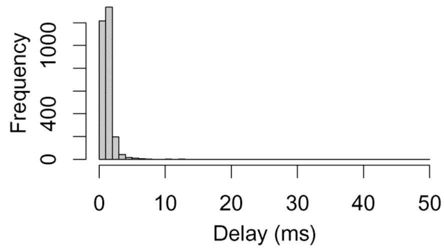
38

DCF, `VoIP` downlink delay, 4 *downloads*



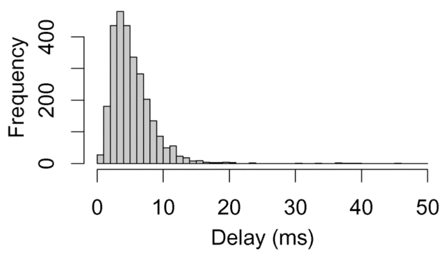
39

DCF, `VoIP` uplink delay, 4 *downloads*



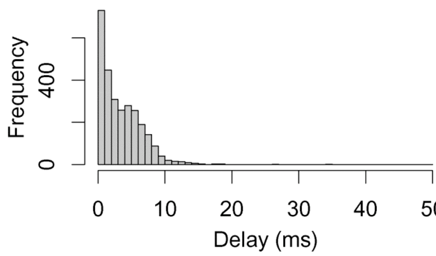
40

AAP, `VoIP` downlink delay, 4 *downloads*



41

AAP, `VoIP` downlink delay, 4 *downloads*



42

Shares

- Measurements of UDP
  - DCF - almost the same share per station:
    - 5509+5489+5297+5561+5825 = 27681 Mb/s
  - AAP (802.11a AAP with CW of 13):
    - 2640+2651+2604+2704+17600 = 28199 Mb/s
  - share of 62.4 % compared to 67% theoretical

Conclusions

- Simple and elegant solution at MAC layer to the unfairness problem
  - right shares of transmission opportunity
  - correct operation of TCP over 802.11 - self-clocked flow control
  - keeps empty buffer at AP - gives short delays
  - always preference to download connections
- Optimal solution in mixed upload/download scenarios requires upper layer modification
  - proper scheduling at IP/TCP layer