Squeezing 100+ VoIP calls out of 802.11b WLANs

Sangki Yun*, Hyogon Kim*, and Inhye Kang+
*Korea University, +University of Seoul
*{lockdown99,hyogon}@korea.ac.kr, +inhye@uos.ac.kr

Abstract

To solve the poor bandwidth efficiency problem of VoIP over IEEE 802.11 wireless LANs (WLANs), the literature frequently suggests elongating voice payload size. Such "voice frame aggregation" can be performed (1) over a single call or multiple calls, (2) at the voice source or at the wireless access point (AP), and (3) with additional repacketization delay or without it. This paper shows that over IEEE 802.11 WLANs, voice frame aggregation should better be done at the AP, over a single call, and without additional delay. We demonstrate through simulation and analysis that this particular implementation choice enables us to support far more VoIP calls on 802.11 WLANs than other choices. For one instance, we show that 100 or more G.729 calls are possible on 802.11b within ITU-T recommended delay bounds.

1. Introduction

We have known for some time that running Voice over IP (VoIP) over 802.11 LANs is extremely inefficient [1-4]. For instance, Table I summarizes the total overhead involved in a single G.729 [5] voice frame transmission with 20ms voice sample, over the 802.11b link [6]. It testifies to the stunning fact that the bandwidth efficiency for VoIP calls in G.729 codec over 802.11b link is bounded by 1.74% even without any other traffic. The G.729 standard of 10ms sample would have a worse result, and other codecs are not much different. It is why the maximum number of sustainable calls on 802.11b link is said to be only on the order of 10s [1], 20s [2], or even less than 10 [3], depending on the voice traffic model and the codec.

A few approaches to improving the efficiency are conceivable: header compression [7,8], frame aggregation [1,9,10], or even MAC protocol modification [11-13]. But about header compression, Table I tells us that the header overhead above the network layer (i.e. IP/UDP/RTP) accounts for only 3.48% of the total, alluding to the marginality of any approach to save on them. And so far, MAC and PHY header compression has not been considered.

The fact that bulk of the overhead lies in the channel access, MAC/PHY framing, 802.11 ACK transmission and inter-frame spaces (IFSs) suggests that reducing the sheer number of MAC transmissions should be the most effective approach. "Frame aggregation" that packs multiple voice samples into a single MAC frame can achieve the very effect. Even in wired networks, frame aggregation for VoIP traffic has been shown to yield some bandwidth savings with a single call [9] and significantly more with multiple calls [10]. In wireless environment, it should be all the more effective due to higher per-frame overheads and the existence of channel access overhead, some of which are listed in Table I. Indeed, most proposals that take on the VoIP inefficiency problem on the 802.11 WLANs follow this path [1-4,14]. In this paper, we will assess these approaches both qualitatively and quantitatively.

TABLE I. TYPICAL OVERHEAD OF A SINGLE G.729 VOICE FRAME TRANSMISSION OVER THE 802.11B LINK

Delay component	Time (µs)	Fraction (%)
DIFS	50	5.98
Average[1] channel access delay due to CA	310	37.11
Voice Frame (G.729)	**14.55**	1.74
RTP/UDP/IP encapsulation	29.09	3.48
LLC/SNAP encapsulation	7.27	0.87
MAC header and trailer	20.36	2.44
Physical-layer (PLCP) preamble and header	192	22.98
SIFS	10	1.20
PLCP preamble and header	192	22.98
MAC header and trailer	10.18	1.22
Total	**835.45**	**100.00**

2. Zero-delay Frame Aggregation

The traditional thought on frame aggregation is that we delay the (re)packetization to gather more frames. But in this paper we offer a novel view on it: *we need to save bandwidth through frame aggregation only when there is bandwidth shortage (i.e., congestion) on the*

[1] This is an average value for 802.11b default configuration of *CWmin*=31 (slots). Upon successive transmission failures, it can grow larger. Note that in this paper we ignore the uninteresting case where only a single wireless station is in a BSS.

802.11 link. In essence, there is not really a desperate need to trigger the frame aggregation until we see multiple voice frames from the same call stack up in the MAC queue due to congestion on the wireless link. As we discussed above, turning on the aggregation all the time incurs fixed delay cost associated with it, which existing proposals all take for granted. On the other hand, the novel approach above aggregates voice frames only as needs arise. Figure 1 depicts the operation of this reactive, congestion-regulated frame aggregation that we call zero-delay frame aggregation (ZFA) in this paper.

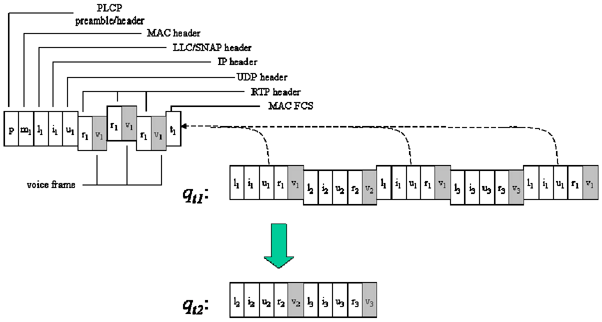

Figure 1. ZFA operation example.

At time t_1, the queue has 5 LLC/SNAP encapsulated voice frames (q_{t1}). When the first voice frame is removed from the queue for MAC transmission attempt, the voice frames from the same call ("1") are coalesced into the same MAC frame. As a consequence, a MAC frame carries out three voice frames from the queue, while the voice frames from other calls ("2" and "3") are left in the queue for later transmissions (q_{t2}). As voice frames are repacketized, they get to share the same UDP and IP header. So some fields in these two headers need to be recomputed, while other headers (LLC/SNAP, MAC, PLCP) are unaffected except that all but one LLC/SNAP headers are dropped. First, UDP checksum and the length fields need to be recomputed. Second, the IP total length and header checksum fields must be correspondingly updated. The IP header checksum needs re-computation because the total length field has changed.

As illustrated in the above example, before the voice frame at the head of the queue is removed for the MAC layer transmission, ZFA inspects the queue for other voice frames from the same call. If there is, ZFA performs the frame aggregation. We remark here that ZFA does *not* incur any frame collection delay since it *does not wait* for the subsequent voice frames. Instead, it aggregates the frames from the same call if they are found residing in the queue concurrently. This is only right, because when there is no congestion, voice frames from the same call will not queue up, and we do not have to worry about the bandwidth efficiency. As congestion sets in, more and more voice frames in the same call will be found concurrently cumulated in the queue, which are cleared altogether when the foremost frame is removed for MAC transmission. Therefore, ZFA self-regulates the aggregation size, automatically adapting to the given load condition on the 802.11 WLAN.

In the rest of the paper, we mainly explore the properties of the ZFA [14] as compared to the traditional source-based intra-call frame aggregation [2-4]. We will not further consider the inter-call aggregation since it comes with prohibitively high implementation complexities in the wireless LAN environment.

3. Analysis

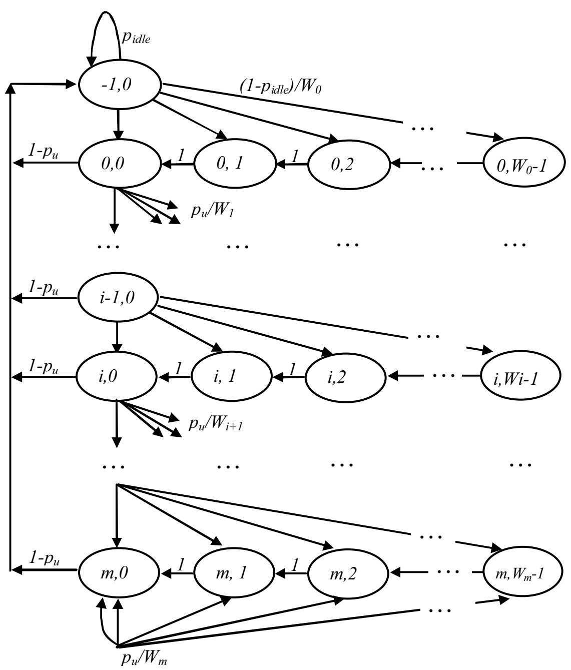

Figure 2. Markov chain model of a wireless station (non-saturated network).

Interestingly, ZFA has a completely different queuing dynamics compared with the traditional source-based frame-aggregated voice flow. This difference turns out to be significant, since it visibly affects the number of VoIP calls that can be accommodated to the 802.11 network. So we provide a brief queuing analysis of ZFA here. Throughout the analysis, we will take G.729 example since it is shown to perform better than G.711 or G.723.1 on 802.11b link [3,4]. As for the voice payload size, we assume 20ms as in Table I. Also, we will assume that VoIP traffic can be distinguished for separate scheduling by using some heuristic method [17], where the prioritized scheduling functionality for voice is already supported by commercial hardware products [16]. To start with, we use constant bit-rate (CBR) traffic as the model of VoIP calls. In the telephony parlance, this means that the voice activity factor α is 1. Although unrealistic, this model helps simplify the analysis and shed light on many notable properties of ZFA. It is fairly straightforward to extend the analysis to a more realistic model such as the ON-OFF model [18] with much smaller voice activity factor [14]. To validate the analysis, we compare it to the simulation, for which we use *ns-2* [19].

The exact delay that a MAC frame takes to cross the 802.11 link can be computed from a Markov chain model that is similar to Bianchi's [20] but taking account of unsaturated operation as shown by Figure 2 [14]. Compared with the Bianchi model, Figure 2 has a new state (-1,0). It represents the idle state in which the station is not attempting transmission. This is because even when we use CBR for VoIP, the stations may not be always backlogged with voice frames. For instance, the standard G.729 codec generates a voice frame every 10ms. With only a few actively transmitting stations, the voice frame could be cleared well before the next voice frame arrives.

Due to the space limit, we omit detailed descriptions of the Markov chain, but in [14], we show the transition probabilities, the transmission probability, the probability that a station stays in idle state and the collision probability for the Markov chain. Also, we will follow the notations of [14].

A. Queuing delay under ZFA

Under ZFA, we will regard the uplink voice frames as experiencing some channel access delay but not any queuing delay. Suppose the head-of-line (HOL) voice frame is v_j upon the arrival of another voice frame v_k to the queue. The queue sojourn time d_k of v_k is bounded by the 802.11 channel access delay d_j of v_j, since as soon as v_j is transmitted so is v_k. So we view d_k as channel access delay instead of queuing delay. This rather peculiar view is just to simplify the analysis of ZFA, and it does not affect any other aspect of the issue. Under this convention, the queuing delay cannot be positive except when there are so many voice frames in the queue so that a single MAC service data unit (MSDU) cannot carry all of them. But with the 802.11 maximum MSDU size of 2304 bytes, it means $d_j > 2304/20 \times 20ms = 2304ms$. Only extremely overloaded network can cause such unacceptable channel access delay, and VoIP calls are impossible due to the QoS problem anyway. So we exclude such case as impractical.

While the voice frames in the same call are aggregated, disparate calls are not transported in the same MAC frame in intra-call aggregation like ZFA. So in the downlink, queuing delay exists. Namely, if the HOL voice frame is not from the same call, a voice frame experiences a larger delay than just the channel access delay of its foremost colleague in the queue. This downlink behavior under ZFA can be approximated by the M/M/1//M model [21], where the customer population (*i.e.*, number of calls) limited to M. Then the expected AP queue length in the steady state is:

$$E[Q^d] = \sum_{k=0}^{M} k \left[\sum_{l=0}^{M} \left(\frac{\lambda}{\mu} \right)^l \frac{l!}{(M-l)!} \right]^{-1} \left(\frac{\lambda}{\mu} \right)^k \frac{k!}{(M-k)!}. \qquad (1)$$

Here, λ denotes the arrival rate with which a station in idle state attempts a transmission of a voice frame. In terms of M/M/1//M, a customer transitions to "arriving" state as soon as its service is completed. λ is the rate at which a customer finally arrives, exiting the arriving state. In ZFA, a call is serviced when its voice frames are shipped out by a MAC frame. Time between this instant and the next voice frame arrival constitutes the sojourn time at the arriving state (*i.e.*, idle state (-1,0) in Figure 2). Since the MAC frame transmission and the voice frame arrival events are completely independent, we assume that the time to the attempt is uniformly distributed. Thus, with T_f =20ms, λ=1/10ms (although not shown for space, we can confirm through experiments that it holds even for lightly loaded network). And the service rate μ is given by:

$$\mu = [E[T_i^d] + E[T_o^d] + E[T_s^d] + cE[T_c^d]]^{-1}, \qquad (2)$$

where $E[T_i]$ is the time spent for backoffs, $E[T_o]$ is the channel time occupied by other nodes, and $E[T_s]$ and $E[T_c]$ are time consumed in a successful transmission and a collision, respectively. And *c=p/(1-p)* is the average number of collisions per transmission attempt. Now, the uplink and the downlink delays are given by

$$E[D^u] = E[T_i^u] + E[T_o^u] + E[T_s^u] + cE[T_c^u], \qquad (3)$$
$$E[D^d] = \left(E[T_i^d] + E[T_o^d] + E[T_s^d] + cE[T_c^d] \right) \times E[Q^d],$$

where $E[Q^d]$ is given by Eq. (1). One caveat here is that the average delays in Eqs. (3) are the delay of the first voice frame v_j in the MAC frame. The voice frames $v_k(k > j)$ aggregated together with v_j experience less delay since they arrived to the queue later. Specifically, they experience $(j-1)\cdot T_s$ less delay than the first frame v_j. So we must modify Eqs. (3) as follows:

$$E[D'] = \sum_{i=0}^{\lfloor E[V] \rfloor} \frac{(E[D] - i \times T_s)}{E[V]} \quad . \qquad (4)$$

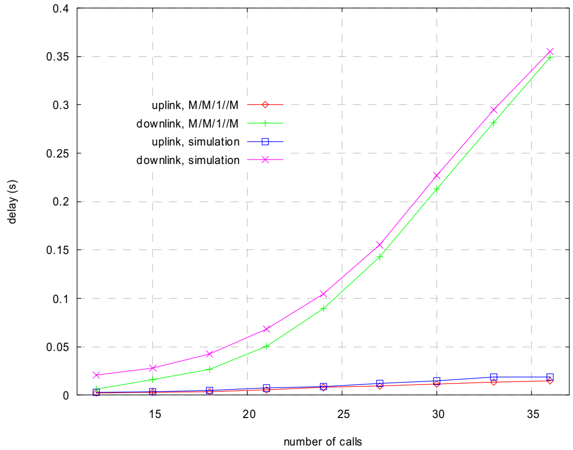

Figure 4. Calls vs. 802.11 link delays on the under ZFA, 20ms payload.

Figure 4 plots the uplink and downlink delays against the number of calls as computed by Eq. (4) and as observed in simulation. If the average delay budget for the 802.11b link $\bar{d}_{req}^{802.11}$ is 100ms, for instance, the number of sustainable calls is around 24 as the downlink hits the limit first. This result with ZFA is approximately 2-fold increase in the number of calls compared to the vanilla 802.11b as we will show below. Still this is far less than we should expect out of the 802.11b capacity: G.729 codec rate is merely 8kbps.

The most important implication from Figure 4 is that there is ample room in which the downlink delay can be reduced through delay redistribution between uplink and downlink. In other words, we can attempt to reduce the delay of the AP by giving more transmission opportunities to the AP, which is amortized by the wireless stations. By doing this, for example, the ZFA downlink delay curve can further tilt to the right.

B. Additional boosting though symmetrization

Since there is hope for ZFA that by giving a larger share of the 802.11 bandwidth to the AP we can boost the number of calls, here we extend our exploration in that direction. Fortunately, it is provable that the bandwidth and delay distribution among stations is precisely controlled through appropriate CW_{min} configuration [35,36]. Specifically, the channel access delay ratio is proportional to CW_{min} ratio. We could achieve the delay redistribution in two ways: either give smaller CW_{min} for AP or give larger CWmin for wireless stations. In this paper, we take the latter approach, i.e., fix the CW_{min} for AP while scaling it for wireless stations in proportion to the VoIP call intensity. In terms of the implementation, the former is simpler – the AP can just adjust its CW_{min} internally. But for the latter, the AP must broadcast the CW_{min} value that the wireless stations should use in the beacon, which the 802.11e standard has as a feature [23]. However, there is a reason that we take the latter approach – congestion control. The increase in the VoIP call volume increases both asymmetry and traffic load. Thus CW_{min} adaptation should serve double purpose, to adjust the delay of uplink and downlink, and to relieve the 802.11 link of the excessive collision probability arising from the increased VoIP traffic. For convenience, we will refer to this optimization technique as Contention Window Adaptation (*CWA*) in the discussion below.

The key idea of CWA can be summarized as follows:

$$CW_{\min}^{(W)} = CW_{\min}^{def} \cdot \gamma \qquad (5)$$

where $CW_{\min}^{def}$ is the default minimum contention window size (i.e., 31 slots in 802.11b) and γ is the effective number of contending stations for the uplink transmission. We use γ instead of the nominal number of stations n since the system is not saturated. Obviously, we can easily obtain this γ by analyzing the Markov chain in Figure 2. In the Markov chain, $b_{-1,0}$ denotes the probability where a station idles because it has not received a voice frame after the previous transmission. So being in the state means that the station is not contending. Then the probability that a station is in a contending state is $1-b_{-1,0}$, and the effective number of contending calls in the uplink is given by $\gamma = n \cdot (1-b_{-1,0})$.

But how can the AP estimate γ in practice? Obviously, it cannot be running the Markov chain. One way is to track the number of actively transmitting stations in the voice class, since the AP is at a good position to do so in the infrastructure mode, *e.g.*, through MAC address examination. (We exclude the case of more than one call concurrently originating from a wireless station as impractical). However, this method is complicated. So, here we propose a simpler and more precise method that does not require separate and constant monitoring. The idea comes from Eqs. (3).

Under the same CW_{min}, T_i, T_s, T_c are all equal in uplink and downlink directions. So, by approximating the uplink delay and the downlink delay without the queue length term $E[Q^d]$, we realize that we need to inflate the uplink terms (i.e., $T_i^u, T_o^u, T_s^u, T_c^u$) by a factor of $E[Q^d]$ to balance delays. In other words, we set $\gamma = E[Q^d]$. This is the quintessence of the idea in estimating γ.

Figure 7 shows the effect of using CWA in addition to ZFA under 20ms payload both by analysis and simulation. Compared with the ZFA alone, we notice that the downlink delay has been drastically reduced in ZFA+CWA at the cost of increased delay in the uplink. According to the simulation, $\overline{d}_{req}^{802.11} = 100ms$ allows 90 calls. This is a significant jump from the ZFA result in Figure 4, not to mention the 7-8 fold increase from the default case (*i.e.*, vanilla 802.11b with 20ms voice payload) as shown in Figure 5. With more lenient delay requirement, the number of calls could well exceed 100.

In essence, the M/M/1//M-like behavior of ZFA implies that it self-regulates the load by reactive aggregation so that the delay does not suddenly jump. It opens a possibility of delay mitigation by shifting it to the uplink channel access delays.

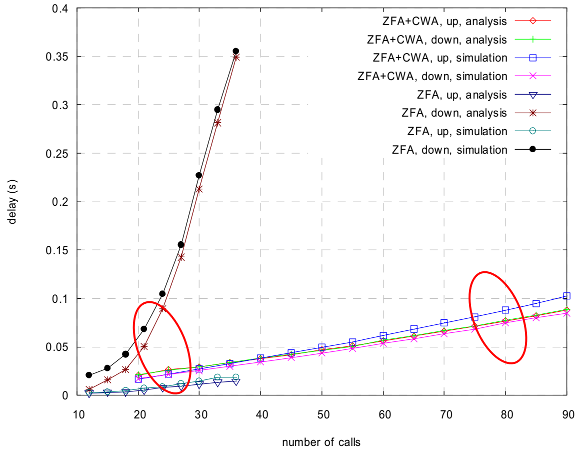

Figure 7. Uplink/downlink delay under ZFA and ZFA+CWA.

C. *Comparison of ZFA+CWA and source-based FA revisited*

Now, we compare the delay characteristics of ZFA (enhanced with CWA) with that of the traditional source-based frame aggregation. Figure 9 compares them as a function of the number of calls. We assume 20ms voice payload for ZFA+CWA and 30, 50, and 80ms voice payload for the source-based aggregation. Although 20ms for ZFA+CWA would also fall into the source-based aggregation category under our classification scheme, many commercial products use 20ms as the default value for G.729 for increased efficiency (*e.g.*, Cisco products [34]), so we also use it as the base case. Uplink delays are not plotted in the figure for clarity but they are consistently very close to 0. With ZFA+CWA, for instance, it is 0.0031 at 15 calls with 20ms payload. Without ZFA+CWA they are also close to zero but only until the downlink delay explodes. As we can clearly see, the downlink delay dynamics is completely different between ZFA+CWA and source-based aggregation. Specifically, with source-based aggregation, the delay becomes suddenly unbounded at a threshold, while under ZFA+CWA it grows far more slowly.

Let us consider the 30ms source-based FA case and figure out why only 17 G.729 calls becomes the threshold in the figures. As we see in Table I, an optimistic 2 average estimate of a voice frame transmission time is 835µs. Since our G.729 voice frames are generated every 30ms, the 18 wireless stations collectively produce 600 voice frames per second in the uplink. And in the downlink direction, corresponding amount of traffic takes place. In this situation, the utilization of the 802.11b link is:

$$\rho = \lambda / \mu = 1200 \times 835.45 \times 10^{-6} = 1.0025 .$$

Therefore, 18 is about the number where we begin to face $\rho > 1$. With 50ms and 80ms payload the threshold comes later (28 calls and 39 calls, respectively), but the dynamics is still the same. As M/M/1//M model would suggest, however, the delay with reactive ZFA less abruptly increases as more calls come in to the system.

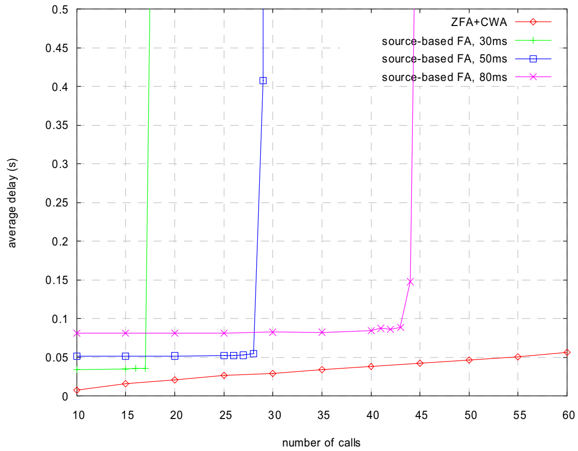

Figure 9. Downlink delay, with 20ms voice payload for ZFA+CWA and 30, 50, and 80ms voice payload for source-based FA.

[2] It is the average channel access delay in the absence of backoffs due to contention, *i.e.*, 310µs.

0-7695-2593-8/06 $20.00 © 2006 IEEE

D. Peak delay consideration

So far, we have considered the *average* delay requirement $\overline{d}_{req}^{802.11b}$ mainly for ease of analysis and comparison with other schemes that use average delay in the estimation of the call capacity. In particular the Hole-Tobagi bound [14] and Garg &Kappes [13] compute the capacity based on the average delay. But in reality, what matters more is the peak delay. Due to the variability caused by queuing and channel access time on the 802.11 link, the peak delay could be much higher than the average. For such variability there is the de-jittering buffer at the receiver. It is designed to absorb the variability in the interarrival times of voice frames, typically up to 2 or 3 times of T_s. But if the jitter of a voice frame is so large that even the buffer cannot handle it, the frame cannot be used to reproduce voice and are simply thrown away. Such "delay losses" affect the call quality, so here we take account of such delayed frames in our estimation of the sustainable calls. And speaking of the realistic number, now we will use the α =0.43 for the voice activity factor.

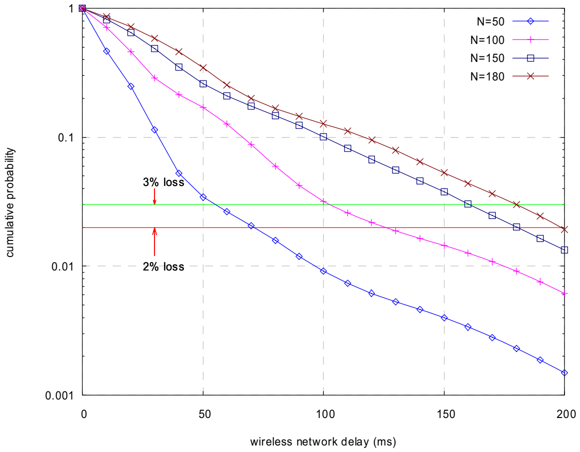

Figure 11. CCDF for aggregated G.729 voice frames.

Figure 11 shows the CCDF of the 802.11b delay experienced by the aggregated G.729 voice frames with different number of calls. According to an ETSI experiment [45], the Mean Opinion Score (MOS) rating for a G.729 call stays above 3.0 ("fair") if the loss rate is maintained below 2 to 3%, where the maximum MOS for G.729 is 3.6. In the figure, we observe that with 3% loss and $\hat{d}_{req}^{802.11b} = 100ms$ peak delay requirement, approximately 100 calls can be supported. Namely, if we require that the absolute delay over the 802.11b link be bounded by 100ms, about 3% of the voice frames are lost due to excessive jitter with N=100. As expected, there is a sharp fall in the number of sustainable calls as a consequence of using peak delay instead of average delay. It is about half of what we could support under $\overline{d}_{req}^{802.11b} = 100ms$. With a more lenient requirement of $\hat{d}_{req}^{802.11b} = 150ms$, however, the sustainable number rebounds to 148.

Table III compares at a glance the number of calls N supported on 802.11b under different proposals considered in the paper, as reported in the literature [1-4,14]. As can be seen, proposals differ in codec, voice payload size, voice activity factor, QoS requirements, or the used methods, so the direct comparison is not entirely straightforward. But except for Wang, the schemes compared with ZFA+CWA are source-based frame aggregation, so Figure 9 should amply tell us the difference in terms of the adaptability and efficiency.

For Hole, the delay requirement includes the elongated RTP packetization delay. So in practice, it is as good as allowing 130 (=100+50-20)ms peak delay for ZFA+CWA. This is because ZFA spends so much less time collecting voice payload at the source. Medepalli must be viewed similarly, so it ought to be compared to ZFA+CWA with 130ms, too. Garg uses the average delay bound, so the number should be compared with that of ZFA+CWA in Figure 10. Moreover, since the 100ms packetization delay is a constant cost, it is equivalent to 180ms average delay requirement for ZFA+CWA. With the peak delay and delay loss requirement of 100ms and 3%, respectively, ZFA+CWA can support up to 85 calls with α =0.43 under 20ms payload (Larger payload size than 20ms will further improve the efficiency, but it will elongate the mouth-to-ear delay as in source-based aggregation.).

TABLE II. COMPARISON OF SUPPORTED CALLS ON 802.11B

Scheme	N	Codec	Requirements		Methods	α
			Delay (ms)	**Loss (%)**		
Wang [11]	17.7	G.711	30	2	M, A	1
	21.7	G.729				
	22	GSM 6.10				
	46					0.43
Hole [4]	34	G.729	150-	0.19	P (50ms)	1
	25	G.711	150-	0		
Garg [13]	66	G.729	100*	1	P (100ms)	1
Medepalli [12]	46	G.711	100	2	P (50ms)	0.48
Baldwin [11]	40	G.711	100	10	C, D	0.43
V100	**100**	**G.729**	100	3	**A, C, S**	**0.43**
	148		150			**0.43**

Delay: *: average delay bound, -: packetization delay included
Methods: M – multicast, A – frame aggregation, P – RTP payload increase, C – contention reduction, S – up/down symmetry enforcement

4. Concluding remarks

This is the first work that demonstrates the number of supported calls on 802.11b link can be boosted to O(100) under good channel condition. The combination of congestion-aware voice frame aggregation and contention window adaptation for uplink-downlink symmetry attains the performance in a highly synergistic manner. Better yet, these mechanisms require no changes in incumbent protocols like 802.11 MAC, RTP, UDP, or IP. Moreover, it does not introduce additional packetization delay as other voice frame aggregation approaches do. Although explained with 802.11b, the proposed idea can be applied to any 802.11 network that needs to create maximal number of calls within the given bandwidth allocation for VoIP.

5. Acknowledgement

This research is supported by the Ubiquitous Autonomic Computing and Network Project, the Ministry of Information and Communication (MIC) 21st Century Frontier R&D Program in Korea.

6. References

[1] W. Wang, S. C. Liew, V. O. K. Li, "Solutions to performance problems in VoIP over a 802.11 wireless LAN," *IEEE Transactions on Vehicular Technology*, 54(1), Jan. 2005.

[2] K. Medepalli, P. Gopalakrishnan, D. Famolari and T. Kodama, "Voice capacity of IEEE 802.11b, 802.11a, and 802.11g wireless LANs," in proceedings of IEEE Globecom 2004.

[3] S. Garg and M. Kappes, "Can I add a VoIP call?," in proceedings of IEEE ICC, 2003.

[4] D. Hole and F. Tobagi, "Capacity of an IEEE 802.11b wireless LAN supporting VoIP," in proceedings of IEEE ICC 2004.

[5] *Coding of speech at 8kbit/s using conjugate-structure algebraic-odeexcited-linear-prediction*, ITU-T Recommendation G.729, Mar. 1996.

[6] ANSI/IEEE, "IEEE 802.11b-1999: Supplement to 802.11-1999,Wireless LAN MAC and PHY specifications: Higher speed Physical Layer (PHY) extension in the 2.4 GHz band," 1999.

[7] S. Casner and V. Jacobson, "Compressing IP/UDP/RTP Headers for Low-Speed Serial Links", *RFC 2508*, February 1999.

[8] L.-E. Johnson and G. Pelletier, "Robust Header Compression (ROHC): A Link Layer Assisted Profile for IP/UDP/RTP," RFC 3242, Apr. 2002.

[9] H. Kim and I. Kang, "Measurement-based Frame Grouping in Internet Telephony, *IEE Electronics Letters*, 37(1) , Jan. 2001.

[10] H. Kim, I. Kang, and E. Hwang, "Measurement-based multi-call voice frame grouping in Internet telephony," *IEEE Communications Letters*, 6(5), May 2002.

[11] R. Baldwin, N. Davis IV, S. Midkiff, and R. Raines, "Packetized voice transmission using RT-MAC, a wireless real-time medium access control protocol," *Mobile Computing and Communications Review*, 5(3), 2001.

[12] T. Hiraguri, T. Ichikawa, M. Iizuka, and M. Morikura, "Novel multiple access protocol for voice over IP in wireless LAN," *IEICE Transactions on Communications*, E85-B(10), Oct. 2002.

[13] A. Banchs, X. Perez, M. Radimirsch, and H. Stuttgen, "Service differentiation extensions for elastic and real-time traffic in 802.11 wireless LAN," in proceedings of IEEE Workshop of High Performance Switching and Routing, May 2001.

[14] S. Yun, H. Kim, and I. Kang, "Squeezing 100+ calls out of IEEE 802.11b wireless LANs," technical report, available at http://widen.korea.ac.kr/v100.pdf.

[15] .Y. Kim, S. Choi, K. Jang, and H. Hwang, "Throughput enhancement of IEEE 802.11 WLAN via frame aggregation," in proceedings of IEEE VTC (Fall), 2004.

[16] SpectraLink Voice Priority: http://www.spectralink.com/products/, 2005.

.

[17] H. Kim, B. Roh, S. Yoo, "Online RTP packet classificantion for real-time multimedia traffic management in the Internet," in proceedings of ICIS, 2002.

[18] P. Brady, "A Model for Generating On-Off Speech Patterns in Two-Way Conversation," Bell Syst. Tech. Journal, 48(7), Sept. 1969.

[19] ns-2, http://www.isi.edu/nsnam/ns.

[20] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE Journal on Selected Area in Comm.*, 18(3), March 2000.

[21] L. Kleinrock, *Queueing Systems*, V.1: Theory, pp. 106-107.

[22] "Voice over IP – per-call bandwidth consumption," Cisco technical note, available at http://www.cisco.com/warp/public/788/pkt-voice-general/bwidth_consume.htm.

[23] ANSI/IEEE, "802.11e: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Medium Access Control (MAC) Enhancements for Quality of Serivce (QoS)," Nov. 2002.