                   On Consensus and Humming in the IETF

Abstract

   The IETF has had a long tradition of doing its technical work through
   a consensus process, taking into account the different views among
   IETF participants and coming to (at least rough) consensus on
   technical matters. In particular, the IETF is supposed not to be run
   by a "majority rule" philosophy. This is why we engage in rituals
   like "humming" instead of voting. However, more and more of our
   actions are now indistinguishable from voting, and quite often we are
   letting the majority win the day without consideration of minority
   concerns. This document explains some features of rough consensus,
   what is not rough consensus, how we have gotten away from it, how we
   might think about it differently, and the things we can do in order
   to really achieve rough consensus.

   Note: This document is quite consciously being put forward as
   Informational. It does not propose to change any IETF processes and
   is therefore not a BCP. It is simply a collection of principles,
   hopefully around which the IETF can come to (at least rough)
   consensus.

Status of This Memo

   This document is not an Internet Standards Track specification; it is
   published for informational purposes.

   This document is a product of the Internet Engineering Task Force
   (IETF). It represents the consensus of the IETF community. It has
   received public review and has been approved for publication by the
   Internet Engineering Steering Group (IESG). Not all documents
   approved by the IESG are a candidate for any level of Internet
   Standard; see Section 2 of RFC 5741.

   Information about the current status of this document, any errata,
   and how to provide feedback on it may be obtained at
   http://www.rfc-editor.org/info/rfc7282.

Copyright Notice

Table of Contents

1. Introduction

   Almost every IETF participant knows the aphorism from Dave Clark's
   1992 plenary presentation [Clark] regarding how we make decisions in
   the IETF:

      We reject: kings, presidents and voting.

      We believe in: rough consensus and running code.

   That is, our credo is that we don't let a single individual dictate
   decisions (a king or president), nor should decisions be made by a
   vote, nor do we want decisions to be made in a vacuum without
   practical experience. Instead, we strive to make our decisions by
   the consent of all participants, though allowing for some dissent
   (rough consensus), and to have the actual products of engineering
   (running code) trump theoretical designs.

   Having full consensus, or unanimity, would be ideal, but we don't
   require it: Requiring full consensus allows a single intransigent
   person who simply keeps saying "No!" to stop the process cold. We
   only require rough consensus: If the chair of a working group
   determines that a technical issue brought forward by an objector has
   been truly considered by the working group, and the working group has
   made an informed decision that the objection has been answered or is
   not enough of a technical problem to prevent moving forward, the
   chair can declare that there is rough consensus to go forward, the
   objection notwithstanding.

   To reinforce that we do not vote, we have also adopted the tradition
   of "humming": When, for example, we have face-to-face meetings and
   the chair of the working group wants to get a "sense of the room",
   instead of a show of hands, sometimes the chair will ask for each
   side to hum on a particular question, either "for" or "against".

   However, in recent years we have seen participants (and even some
   folks in IETF leadership) who do not understand some of the
   subtleties of consensus-based decision making. Participants ask,
   "Why don't we just vote? Why are we bothering with this 'humming'
   thing?" Or even more concerning, "We've already hummed/voted, so why
   isn't the discussion concluded?" Chairs, many of whom have little
   experience in leading large volunteer groups like those in the IETF,
   let alone experience in how to gather consensus, are faced with
   factious working groups with polarized viewpoints and long-running
   unresolved issues that return again and again to the agenda. More
   and more frequently, people walk away from working groups, thinking
   that "consensus" has created a document with horrible compromises to
   satisfy everyone's pet peeve instead of doing "the right thing".

None of these things are indicators of a rough consensus process
being used, and the fact that we are seeing them is likely due to
some basic misperceptions.

This document explains some features of rough consensus, explains
what is not rough consensus, discusses some new ways to think about
rough consensus, and suggests ways that we might achieve rough
consensus and judge it in the IETF. Though this document describes
some behaviors of working groups and chairs, it does so in broad
brushstrokes and it does not prescribe specific procedures. Rather,
this document is intended to foster understanding of the underlying
principles of IETF consensus processes. While it may be of general
interest to anyone interested in the IETF consensus processes, the
primary audience for this document is those who have experience
working in the IETF and are trying to understand and participate in
the consensus-building process, and it is particularly aimed at
generating thought and discussion among those who might lead a
consensus discussion. Although most of the examples in this document
talk about working group chairs, these principles apply to any person
who is trying to lead a group to rough consensus, whether a chair, a
design team leader, a document editor, an area director, or any
community member who is facilitating a discussion or trying to assess
consensus.

While the community has come to rough consensus that the principles
expressed in this document are (at least approximately) right, many
of our current practices are not consistent with these principles.
Again, this document is primarily intended to generate thought and
discussion, not dictate practices. If the IETF does commit itself to
these principles, practices may change in the future.

2. Lack of disagreement is more important than agreement

A working group comes to a technical question of whether to use
format A or format B for a particular data structure. The chair
notices that a number of experienced people think format A is a good
choice. The chair asks on the mailing list, "Is everyone OK with
format A?" Inevitably, a number of people object to format A for one
or another technical reason. The chair then says, "It sounds like we
don't have consensus to use format A. Is everyone OK with format B?"
This time even more people object to format B, on different technical
grounds. The chair, not having agreement on either format A or
format B, is left perplexed, thinking the working group has
deadlocked.

The problem that the chair got themselves into was thinking that what
they were searching for was agreement. "After all", thinks the
chair, "consensus is a matter of getting everyone to agree, so asking

whether everyone agrees is what the chair ought to do. And if lots
of people disagree, there's no consensus." But _determining_
consensus and _coming to_ consensus are different things than
having consensus.

The distinction might be a bit subtle, but it's important.
Engineering always involves a set of tradeoffs. It is almost certain
that any time engineering choices need to be made, there will be
options that appeal to some people, but are not appealing to some
others. In determining consensus, the key is to separate those
choices that are simply unappealing from those that are truly
problematic. If at the end of the discussion some people have not
gotten the choice that they prefer, but they have become convinced
that the chosen solution is acceptable, albeit less appealing, they
have still come to consensus. Consensus doesn't require that
everyone is happy and agrees that the chosen solution is the best
one. Consensus is when everyone is sufficiently satisfied with the
chosen solution, such that they no longer have specific objections to
it.

So, in the case of a working group decision, after the initial
discussion of the pros and cons of the available choices, it is most
important to ask not just for objections to a particular proposal,
but for the nature of those objections. A chair who asks, "Is
everyone OK with choice A?" is going to get objections. But a chair
who asks, "Can anyone not live with choice A?" is more likely to only
hear from folks who think that choice A is impossible to engineer
given some constraints. Following up with, "What are the reasons you
object to choice A?" is also essential. Then, the purported failings
of the choice can be examined by the working group. The objector
might convince the rest of the group that the objections are valid
and the working group might choose a different path. Conversely, the
working group might convince the objector that the concerns can be
addressed, or that the choice is simply unappealing (i.e., something
the objector can "live with") and not a show-stopper. In any event,
closure is much more likely to be achieved quickly by asking for and
trying to accommodate the objections rather than asking for
agreement.

The above discussion does not mean that sorting out disagreements is
the only thing that needs to be done for successful consensus. An
engineering solution that has no objections, but also has no base of
support and is met with complete apathy, is not a solution that has
any useful sort of consensus. Consensus does require the active
engagement and eventual support of those who are working on the
solution. However, finding mere "agreement" among participants is
not enough. People might very well agree that a solution is

sufficient and have no objection to it, but if they also don't
actively think it's a good and correct outcome, it's absurd to
declare that the group has consensus.

There is also an important point to be made about reaching consensus
and "compromising": Unfortunately, the word "compromise" gets used in
two different ways, and though one sort of compromising to come to
consensus is good (and important), the other sort of compromising in
order to achieve consensus can actually be harmful. As mentioned
earlier, engineering always involves balancing tradeoffs, and
figuring out whether one engineering decision makes more sense on
balance compared to another involves making engineering
"compromises": We might have to compromise processor speed for lower
power consumption, or compromise throughput for congestion
resistance. Those sorts of compromises are among engineering
choices, and they are expected and essential. We always want to be
weighing tradeoffs and collectively choosing the set that best meets
the full set of requirements.

However, there is another sense of "compromise" that involves
compromising between people, not engineering principles. For
example, a minority of a group might object to a particular proposal,
and even after discussion still think the proposal is deeply
problematic, but decide that they don't have the energy to argue
against it and say, "Forget it, do what you want". That surely can
be called a compromise, but a chair might mistakenly take this to
mean that they agree, and have therefore come to consensus. But
really all that they've done is capitulated; they've simply given up
by trying to appease the others. That's not coming to consensus;
there still exists an outstanding unaddressed objection. Again, if
the objection is only that the choice is not ideal but is otherwise
acceptable, such a compromise is fine. But conceding when there is a
real outstanding technical objection is not coming to consensus.

Even worse is the "horse-trading" sort of compromise: "I object to
your proposal for such-and-so reasons. You object to my proposal for
this-and-that reason. Neither of us agree. If you stop objecting to
my proposal, I'll stop objecting to your proposal and we'll put them
both in." That again results in an "agreement" of sorts, but instead
of just one outstanding unaddressed issue, this sort of compromise
results in two, again ignoring them for the sake of expedience.

These sorts of "capitulation" or "horse-trading" compromises have no
place in consensus decision making. In each case, a chair who looks
for "agreement" might find it in these examples because it appears
that people have "agreed". But answering technical disagreements is
what is needed to achieve consensus, sometimes even when the people
who stated the disagreements no longer wish to discuss them.

Coming to consensus is when everyone (including the person making the
objection) comes to the conclusion that either the objections are
valid, and therefore make a change to address the objection, or that
the objection was not really a matter of importance, but merely a
matter of taste. Of course, coming to full consensus like that does
not always happen. That's why in the IETF, we talk about "rough
consensus".

3. Rough consensus is achieved when all issues are addressed, but not
    necessarily accommodated

The preceding discussion gives an example where the working group
comes to consensus on a point: Either the objector is satisfied with
the answer to the objection, or the working group is satisfied that
the objection is valid and changes course. But that doesn't happen
all of the time, and it's certainly not the problematic case. Again,
engineering is always a set of tradeoffs. Often, a working group
will encounter an objection where everyone understands the issue and
acknowledges that it is a real shortcoming in the proposed solution,
but the vast majority of the working group believes that
accommodating the objection is not worth the tradeoff of fixing the
problem.

So, an objector might say, "The proposal to go with protocol X is
much more complicated than going with protocol Y. Protocol Y is a
much more elegant and clean solution, which I can code much more
easily, and protocol X is a hack." The working group might consider
this input, and someone might respond, "But we have a great deal of
code already written that is similar to protocol X. While I agree
that protocol Y is more elegant, the risks to interoperability with
an untested solution are not worth it compared to the advantages of
going with the well-understood protocol X." If the chair finds, in
their technical judgement, that the issue has truly been considered,
and that the vast majority of the working group has come to the
conclusion that the tradeoff is worth making, even in the face of
continued objection from the person(s) who raised the issue, the
chair can declare that the group has come to rough consensus. (And
even though this is framed in terms of a "vast majority", even that
is not necessarily true. This point is discussed in more detail in
Sections 6 and 7.)

Note that this portrays rough consensus as a fallback. In one sense,
it is: As a working group does its work and makes its choices, it
behaves as if it is striving toward full consensus and tries to get
all issues addressed to the satisfaction of everyone in the group,
even those who originally held objections. It treats rough consensus
as a sort of "exception processing", to deal with cases where the
person objecting still feels strongly that their objection is valid

and must be accommodated. But it is certainly true that, more often
than not in the IETF, at least someone in the group will be
unsatisfied with a particular decision. In that sense, rough
consensus might be closer to the norm than the exception. However,
when a participant says, "That's not my favorite solution, but I can
live with it; I'm satisfied that we've made a reasonable choice",
that participant is not in the "rough" part of a rough consensus; the
group actually reached consensus if that person is satisfied with the
outcome. It's when the chair has to declare that an unsatisfied
person still has an open issue, but that the group has truly answered
the objection, that the consensus is only rough.

Now, a conclusion of having only rough consensus relies heavily on
the good judgement of the consensus caller. The group must truly
consider and weigh an issue before the objection can be dismissed as
being "in the rough". ("In the rough" is terminology from golf.
"The rough" is the term for the longer grass at the side of the
fairway, and if your ball has landed in the rough you are off course
and away from the normal direction of play. The phrase gets used
quite a bit in the IETF as a play on words to complement "rough
consensus" meaning that you are "in the rough" if you find yourself
not agreeing with the rough consensus.) The chair of a working group
who is about to find that there is only rough consensus is going to
have to decide that not only has the working group taken the
objection seriously, but that it has fully examined the ramifications
of not making a change to accommodate it, and that the outcome does
not constitute a failure to meet the technical requirements of the
work. In order to do this, the chair will need to have a good idea
of the purpose and architecture of the work being done, perhaps
referring to the charter of the working group or a previously
published requirements document, or even consulting with other
experts on the topic, and then the chair will use their own technical
judgement to make sure that the solution meets those requirements.
It is possible that the chair can come to the wrong conclusion, and
the chair's conclusion is always appealable should that occur, but
the chair must use their judgement in these cases. What can't happen
is that the chair bases their decision solely on hearing a large
number of voices simply saying, "The objection isn't valid." That
would simply be to take a vote. A valid justification needs to me
made.

It is important to recognize that this view of rough consensus is a
change from the way it sometimes has been characterized in the IETF.
RFC 1603 [RFC1603] described rough consensus as the "dominant view"
of the group:

> Working groups make decisions through a "rough consensus" process.
> IETF consensus does not require that all participants agree
> although this is, of course, preferred. In general the dominant
> view of the working group shall prevail. (However, it must be
> noted that "dominance" is not to be determined on the basis of
> volume or persistence, but rather a more general sense of
> agreement.) Consensus can be determined by balloting, humming, or
> any other means on which the WG agrees (by rough consensus, of
> course).

The above says that consensus can be "determined" by balloting and
humming, and there are certainly IETF folks who have thought of rough
consensus as being primarily about the percentage of people who agree
with a decision. Indeed, RFC 2418 [RFC2418] adds on to the above
text by stating, "Note that 51% of the working group does not qualify
as 'rough consensus' and 99% is better than rough." This document
actually disagrees with the idea that simply balloting or otherwise
looking at percentages can "determine" consensus. While counting
heads might give a good guess as to what the rough consensus will be,
doing so can allow important minority views to get lost in the noise.
One of the strengths of a consensus model is that minority views are
addressed, and using a rough consensus model should not take away
from that. That is why this document talks a great deal about
looking at open issues rather than just counting the number of people
who do or do not support any given issue. Doing so has some
interesting and surprising implications that are discussed in
subsequent sections.

Any finding of rough consensus needs, at some level, to provide a
reasoned explanation to the person(s) raising the issue of why their
concern is not going to be accommodated. A good outcome is for the
objector to understand the decision taken and accept the outcome,
even though their particular issue is not being accommodated in the
final product.

Remember, if the objector feels that the issue is so essential that
it must be attended to, they always have the option to file an
appeal. A technical error is always a valid basis for an appeal.
The chair in making the consensus call (or whoever is responsible to
hear an appeal) may determine that the issue was addressed and
understood, but they also have the freedom and the responsibility to
say, "The group did not take this technical issue into proper
account" when appropriate. Simply having a large majority of people

agreeing to dismiss an objection is not enough to claim there is
rough consensus; the group must have honestly considered the
objection and evaluated that other issues weighed sufficiently
against it. Failure to do that reasoning and evaluating means that
there is no true consensus.

4. Humming should be the start of a conversation, not the end

We don't vote in the IETF. In some ways, we can't vote: Since the
IETF is not a membership organization, it's nearly impossible to
figure out who would get a vote for any given question. We can't
know who the "members" of any given working group would be at any one
time, and we certainly can't know who all of the "members" of the
IETF would be: That's why we refer to "participants" in the IETF; the
IETF doesn't really have "members". Indeed, we often recruit
additional implementers and other experts into working groups in
order to ensure that broader views are brought into the discussion.
So, voting is simply not practical. We've also decided that coming
to consensus (albeit sometimes rough consensus) is an important thing
to do. Final decisions are supposed to be taken on the mailing list,
which reinforces the idea that we come to consensus by looking at the
open issues and not counting heads. We do, on occasion, take
informal polls to get a sense of the direction of the discussion, but
we try not to treat a poll as a vote that decides the issue. When we
do discuss things face-to-face, we don't want to vote there either;
we want to show that we are coming to consensus. So, sometimes, to
reinforce the notion that we're not voting, instead of a show of
hands, we often "hum".

However, more and more we see people who think that a hum is a sort
of anonymous vote, with some chairs calling every question they have
for the working group by asking for a hum and judging the result by
the loudest hum, even saying things like, "There were lots of hums
for choice 1 and very few hums for choice 2, so it sounds like we
have rough consensus for choice 1." This misses some really
important points of using humming and is almost certainly mis-
assessing the consensus. Hums should not be used as votes.

So, why should we engage in this strange practice of humming? What
are good reasons to "take a hum"? One reason is pragmatic. Quite
often, a chair is faced with a room full of people who seem to be
diametrically opposed on some choice facing the group. In order to
find a starting place for the conversation, it can be useful for the
chair to ask for a hum to see if one of the choices already has a
stronger base of support than the other (or any significant base of
support at all, for that matter). Sometimes the hum can tell a chair

that the room isn't all that contentious after all, that it's just a
few voices who were being especially vociferous during the initial
discussion.

Sometimes, the hum will make it clear that choice "foo" has a
significant amount more support than choice "bar", and it is
therefore likely easier to start the discussion by saying, "OK, 'foo'
seems to have quite a bit of support. Let's have the people that
think 'foo' is a bad idea come up and tell us why it is problematic."
At that point, the group can start going through the issues and see
if any of them are showstoppers. It could always turn out that one
of the objections is instantly recognized by the entire group as a
fatal flaw in "foo" and the group will then turn to a discussion of
the merits (and demerits) of "bar" instead. All that the hum does is
give the chair a starting point: The hum indicated that there were
less objections to "foo" than to "bar" at the beginning of the
discussion, so starting with the objections to "foo" might shorten
the discussion.

Another good reason for us to hum is because it actually gives the
chair the opportunity to take the temperature of the room. A smaller
bunch of loud hums for choice A and a larger number of non-committal
hums for choice B might indicate that some people believe that there
are serious problems with choice B, albeit the more popular by sheer
number of people. The chair might decide that starting with choice A
and getting objections to it is the easier path forward and more
likely to result in consensus in the end. Remember, coming to
consensus is a matter of eliminating disagreements, so the chair
wants to choose the path that gets to the least objections fastest.
A bunch of people who are not strongly committed to B might have no
real technical objection to A, even though it is not their first
preference. There is always a chance that this could be misleading,
or even abused, because some people are more willing to hum loudly
than others (just by dint of personality), or that one of the quieter
hums actually turns out to be a show-stopper that makes the original
choice impossible. However, keep in mind that taking the hum in this
case is to figure out how to start the conversation. The chair could
always be surprised because the hum turns out to be unanimous and no
further discussion is needed. Otherwise, the hum begins the
discussion, it doesn't end it.

But couldn't all of the above could have been done with a show of
hands instead of a hum? Absolutely. Indeed, on a mailing list there
is no way to use humming and so a different kind of polling would be
needed. Even in face-to-face situations, sometimes we do use a show
of hands. But there are more symbolic reasons for using a hum
instead of a show of hands when face-to-face: Of course, a chair
could get the temperature of the room by doing a show of hands too,

and knowing who specifically feels one way or another can help a good
chair guide the subsequent conversation. However, a show of hands
might leave the impression that the number of people matters in some
formal way. A chair and a working group with a solid understanding
of how consensus works can certainly do a show of hands and achieve
exactly the same result as a hum. But with less experienced folks, a
show of hands can end up reinforcing the mistaken notion that a vote
is taking place. A chair can always take the hum and then later ask
for specific folks to identify themselves to elicit more discussion.
The advantage of the hum is that it makes it perfectly clear that the
chair is simply figuring out the direction of the conversation.

This also points to another misuse of any kind of informal polling:
If the chair is already convinced that the group has come to
consensus, there isn't much reason to take a poll. In fact, taking a
poll can serve to discourage those who might be in the minority from
voicing their concerns to the group in the face of a large majority
who wants to move forward. Often, the right thing for the chair to
do if they already sense consensus is to say, "It sounds to me like
we have consensus for choice A. Does anybody have any concerns about
or objections to going with A?" This allows folks to bring up issues
to the group that the chair might have mistakenly missed without
having them feel that the majority has "already spoken".

The reverse situation can also have similar advantages and
disadvantages: Sometimes a chair (say, of a birds-of-a-feather
session, or a working group discussing a new proposed document) might
want to make sure that there really is a good base of support to go
forward with a proposal, and takes a hum. This can let the chair see
if there are more than a handful of active people who are really
interested in the new work. However, this has pitfalls as well:
Someone may be dissuaded from raising what could be an essential
concern if they feel that the group is overwhelmingly in favor of
going forward, or conversely some folks may decide to "hum along with
the crowd" even though they're not committed to the outcome. Indeed,
the formulation, or even the order, of questions asked during a hum
can have huge effects on the outcome: Asking simply, "Who supports
going forward with this proposal?", and asking it first, can itself
cause more people to hum in the affirmative than would for
differently formulated questions, or asking the same question after
some more "negatively" framed questions. Any sort of polling,
whether hums or even a show of hands, must be done with caution and
should almost always be used to prompt discussion and questions, not
to conclude the matter.

There are times where the result of a hum is a pretty even split. In
practical terms, that means it doesn't matter where the chair starts
the discussion. And in fact, we've had working groups where a coin

flip decided which proposal to start with. That doesn't mean that
the coin flip determined the outcome; if a fatal technical flaw was
found in the solution that won the coin flip, it is still incumbent
upon the group to address the issue raised or abandon that solution
and find another. Rough consensus on the technical points, in the
end, is always required. Any way to find a place to start, be it the
hum or the coin flip, is only getting to the beginning of the
discussion, not the end.

5. Consensus is the path, not the destination

   We don't try to reach consensus in the IETF as an end in itself. We
   use consensus-building as a tool to get to the best technical (and
   sometimes procedural) outcome when we make decisions. Experience has
   shown us that traditional voting leads to gaming of the system,
   "compromises" of the wrong sort as described earlier, important
   minority views being ignored, and, in the end, worse technical
   outcomes.

   Coming to consensus by looking for objections, tracking open issues,
   and using hums as the start of discussions and not the end can all
   take some patience. Indeed, sometimes objection-based or issue-based
   decision making can be extremely difficult because there can be large
   factions who have diametrically opposed views. And there is no doubt
   that we do see some amount of political compromise (that is, the
   undesirable kind of compromise) from time to time in the IETF.

   However, accepting these things has its price. When we decide that a
   discussion is too factious and opt to simply go with a majority, it
   creates more polarized arguments in the future: Instead of working
   toward the best technical outcome that most everyone can accept,
   people are much quicker to run to opposing sides and dig in to their
   positions. And when we allow real technical issues to drop because
   proponents have simply capitulated or have "horse-traded" to allow
   other technical problems to remain, the end product is weaker.
   Though the IETF can never be perfectly principled with regard to
   rough consensus, failing to be vigilant about sticking to the
   principles makes it increasingly hard to stick to them in the future,
   and ends us up with worse technical output.

   Again, coming to consensus is not the goal in itself. Coming to
   consensus is what we do during our processes to arrive at the best
   solution. In particular, "declaring" consensus is not an end goal.
   Attempts to declare consensus at the end of a discussion just for the
   sake of being able to say that there is consensus often get us back
   into the voting mentality that we're trying to avoid.

We often hear chairs say that they are making a "consensus call".
Sometimes, they simply mean they are making a call _of_ the
consensus; that is, they are declaring the consensus that has, in
their view, been reached when the discussion has reached an end.
That's a fine thing and what chairs are supposed to do: They are
"calling" the consensus. Sometimes, when a chair says that they are
making a "consensus call", the chair is actually making a call _for
discussion_ of a particular point in order to reach consensus.
Although it's a bit odd to call that a "consensus call" (as opposed
to a "call for discussion" or the like), it is fine for a chair to
occasionally identify a particular point of contention and get the
group to focus discussion on it in order to reach consensus. But
more and more often, we hear chairs say that they are making a
"consensus call" at the end of a discussion, where the chair will
pose the classic "Who is in favor of choice A? Who is in favor of
choice B?" questions to the working group. That's not really a
"consensus call", and has the same potential problems as the "hum" at
the end of a discussion: It can be tantamount to asking for a vote.
Even talk of "confirming consensus" has this problem: It implies that
you can confirm that there is consensus by counting people, not
issues. The important thing for a chair to do is to "call consensus"
in the sense of declaring the consensus; others can always object and
say that the chair has gotten the consensus wrong and ask for
reconsideration. However, the chair ought to be looking for
consensus throughout the discussion, not asking for it at the end.

There are some times where chairs will ask a question or take a poll
toward the end of a discussion in order to figure out the state of
consensus, but this must be done with extreme caution. This is
discussed in the next section.

6. One hundred people for and five people against might not be rough
    consensus

    Section 3 discussed the idea of consensus being achieved when
    objections had been addressed (that is, properly considered, and
    accommodated if necessary). Because of this, using rough consensus
    avoids a major pitfall of a straight vote: If there is a minority of
    folks who have a valid technical objection, that objection must be
    dealt with before consensus can be declared. This also reveals one
    of the great strengths of using consensus over voting: It isn't
    possible to use "vote stuffing" (simply recruiting a large number of
    people to support a particular side, even people who have never
    participated in a working group or the IETF at all) to change the
    outcome of a consensus call. As long as the chair is looking for
    outstanding technical objections and not counting heads, vote
    stuffing shouldn't affect the outcome of the consensus call.

So, in a large working group with over 100 active participants and
broad agreement to go forward with a particular protocol, if a few
participants say, "This protocol is going to cause congestion on the
network, and it has no mechanism to back off when congestion occurs;
we object to going forward without such a mechanism in place", and
the objection is met with silence on the mailing list, there is no
consensus. Even if the working group chair makes a working group
"last call" on the document, and 100 people actively reply and say,
"This document is ready to go forward", if the open issue hasn't been
addressed, there's still no consensus, not even rough consensus.
It's the existence of the unaddressed open issue, not the number of
people, which is determinative in judging consensus. As discussed
earlier, you can have rough consensus with issues that have been
purposely dismissed, but not ones that have been ignored.

This brings us back to when a poll could be used (cautiously) at the
end of a discussion. Let's say a discussion has been ongoing for
some time, and a particular objection seems to be holding up the
decision. A diligent chair who's been carefully listening to the
discussion might think, "I have heard person X make this objection,
and I've heard responses from many other folks that really address
the issue. I think we have rough consensus. But the objection keeps
coming up. Maybe it's just the one person getting up again and again
with the same argument, but maybe we don't have rough consensus. I'm
not sure." At this point, the chair might ask for a hum. If only a
single hum objecting can be heard, even a loud one, in the face of
everyone else humming that the objection has been answered, the chair
has pretty good reason to believe that they heard the single
objection all along and it really has been addressed. However, to
say immediately after the hum, "It sounds like we have rough
consensus" and nothing else is at best being slipshod: What the chair
really needs to say at that point is, "I believe the only objection
we've heard is A (coming from person X), and I've heard answers from
the group that fully address that issue. So, unless I hear a
different objection than the one I've just described, I find that
there is rough consensus to move on." That leaves the door open for
someone to tell the chair that the objection was really on different
grounds and they misevaluated, but it makes it clear that the chair
has found rough consensus due to the discussion, not due to the hum.
Again, it's not the hum that ends things, it's that the issues have
been addressed. If the small minority (even one person) still has an
issue that hasn't been addressed, rough consensus still hasn't been
achieved.

Even if no particular person is still standing up for an issue, that
doesn't mean an issue can be ignored. As discussed earlier, simple
capitulation on an issue is not coming to consensus. But even in a
case where someone who is not an active participant, who might not

care much about the fate of the work, raises a substantive issue and
subsequently disappears, the issue needs to be addressed before the
chair can claim that rough consensus exists.

7. Five people for and one hundred people against might still be rough
    consensus

    This one is the real mind-bender for most people, and certainly the
    most controversial. Say there is a very small working group, one
    with half a dozen truly active participants who are experts in the
    field; everybody else is just following along but not contributing to
    the discussion. The active folks come up with a protocol document
    that they all agree is the right way forward, and people inside and
    outside the working group agree that the protocol is likely to get
    widespread adoption; it is a good solution to a real problem, even if
    the non-experts don't have the ability to fully judge the details.

    However, one of the active members has an objection to a particular
    section: The protocol currently uses a well-known algorithm to
    address an issue, but the objector has a very elegant algorithm to
    address the issue, one which works especially well on their
    particular piece of hardware. There is some discussion, and all of
    the other contributors say, "Yes, that is elegant, but what we're
    using now is well-understood, widely implemented, and it works
    perfectly acceptably, even on the objector's hardware. There is
    always some inherent risk to go with a new, albeit more elegant,
    algorithm. We should stick to the one we've got." The chair follows
    the conversation and says, "It sounds like the issue has been
    addressed and there's consensus to stick with the current solution."
    The objector is not satisfied, maybe even saying, "But this is silly.
    You've seen that my algorithm works. We should go with that." The
    chair makes the judgement that the consensus is rough, in that there
    is still an objector, but the issue has been addressed and the risk
    argument has won the day. The chair makes a working group last call.

    Then, the worst-case scenario happens. The objector, still unhappy
    that their preferred solution was not chosen, recruits one hundred
    people, maybe a few who were silent participants in the working group
    already, but mostly people who work at the same company as the
    objector and who never participated before. The objector gets them
    all to post a message to the list saying, "I believe we should go
    with the new elegant algorithm in section Z instead of the current
    one. It is more elegant, and works better on our hardware." The
    chair sees these dozens of messages coming in and posts a query to
    each of them: "We discussed this on the list, and we seemed to have
    consensus that, given the inherent risk of a new algorithm, and the
    widespread deployment of this current one, it's better to stick with
    the current one. Do you have further information that indicates

something different?" And in reply the chair gets utter silence.
These posters to the list (say, some of whom were from the company
sales and marketing department) thought that they were simply voting
and have no answer to give. At that point, it is within bounds for
the chair to say, "We have objections, but the objections have been
sufficiently answered, and the objectors seem uninterested in
participating in the discussion. Albeit rough in the extreme, there
is rough consensus to go with the current solution."

Though the above example uses the most extreme form of recruiting
sheer numbers of people (i.e., from the sales and marketing
department), the same principle should hold true no matter how new or
how credible the objectors seem: The chair is trying to discover
whether objections have been addressed or if there are still open
issues. If, instead of a bunch of sales and marketing people, the
new people to the conversation are developers or others who are
directly involved in creating the technology, or even folks who have
been participating the entire time whose knowledge of the technology
is not in question at all, the principle is still the same: If the
objection has been addressed, and the new voices are not giving
informed responses to that point, they can still justifiably be
called "in the rough". Of course, the more involved and knowledgable
the objectors are, the more difficult it will be for the consensus
caller to make the call, but a call of rough consensus is reasonable.
The chair in this case needs to understand what the responses mean;
only sufficiently well-informed responses that justify the position
taken can really "count".

There is no doubt that this is the degenerate case and a clear
indication of something pathological. But, this is precisely what
rough consensus is ideally suited to guard against: vote stuffing.
In the presence of an objection, the chair can use their technical
judgement to decide that the objection has been answered by the group
and that rough consensus overrides the objection. Now, the case
described here is probably the hardest call for the chair to make
(how many of us are willing to make the call that the vast majority
of people in the room are simply stonewalling, not trying to come to
consensus?), and, if appealed, it would be incredibly difficult for
the appeals body to sort out. Indeed, it is likely that if a working
group got this dysfunctional, it would put the whole concept of
coming to rough consensus at risk. But still, the correct outcome in
this case is to look at the very weak signal against the huge
background noise in order to find the rough consensus.

8. Conclusion

   Although this document talks quite a bit about the things chairs,
   working groups, and other IETF participants might do to achieve rough
   consensus, this document is not really about process and procedures.
   It describes a way of thinking about how we make our decisions.
   Sometimes, a show of hands can be useful; sometimes, it can be quite
   damaging and result in terrible decisions. Sometimes, using a device
   like a "hum" can avoid those pitfalls; sometimes, it is just a poorly
   disguised vote. The point of this document is to get all of us to
   think about how we are coming to decisions in the IETF so that we
   avoid the dangers of "majority rule" and actually get to rough
   consensus decisions with the best technical outcomes.

9. Security Considerations

   "He who defends with love will be secure." -- Lao Tzu

10. Informative References

   [Clark]   Clark, D., "A Cloudy Crystal Ball - Visions of the
              Future", Proceedings of the Twenty-Fourth Internet
              Engineering Task Force, pages 539-543, July 1992,
              <http://www.ietf.org/proceedings/24.pdf>.

   [RFC1603] Huizer, E. and D. Crocker, "IETF Working Group Guidelines
              and Procedures", RFC 1603, March 1994.

   [RFC2418] Bradner, S., "IETF Working Group Guidelines and
              Procedures", BCP 25, RFC 2418, September 1998.

   [Sheeran] Sheeran, M., "Beyond Majority Rule: Voteless Decisions in
              the Religious Society of Friends", ISBN 978-0941308045,
              December 1983.

Appendix A. Acknowledgements

   This document is the result of conversations with many IETF
   participants, too many to name individually. I greatly appreciate
   all of the discussions and guidance. I do want to extend special
   thanks to Peter Saint-Andre, who sat me down and pushed me to start
   writing, and to Melinda Shore for pointing me to "Beyond Majority
   Rule" [Sheeran], which inspired some of the thinking in this
   document.

Author's Address

   Pete Resnick
   Qualcomm Technologies, Inc.
   5775 Morehouse Drive
   San Diego, CA 92121
   US

   Phone: +1 858 651 4478
   EMail: presnick@qti.qualcomm.com