

# RFC 7282 : On Consensus and Humming in the IETF

Stéphane Bortzmeyer

<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 1 juillet 2014

Date de publication du RFC : Juin 2014

<https://www.bortzmeyer.org/7282.html>

---

L'IETF, organisme de normalisation des protocoles de l'Internet, fonctionne normalement par « consensus ». Il n'y a pas de vote formel, pas de majorité qui décide, on discute jusqu'à ce que presque tout le monde soit d'accord. À cause du « presque », on parle de « consensus approximatif » (*"rough consensus"*). Ça, c'est la théorie. En pratique, les évolutions récentes font que de plus en plus de décisions sont prises suite à des opérations qui ressemblent pas mal à des votes majoritaires, justement ce que les règles de l'IETF voulaient éviter. D'où ce RFC d'opinion, écrit pour défendre le principe du consensus approximatif, et pour appeler à se méfier de ces dérives.

Ce RFC a le statut « pour information » et représente le point de vue de son auteur : ce n'est pas un document officiel. En outre, il ne définit pas de nouvelles règles, il explique juste le fonctionnement idéal. Pour résumer ce RFC, revenons à la théorie. Le fonctionnement normal de l'IETF a été traduit par une percutante phrase de Dave Clark (dans « *"A Cloudy Crystal Ball - Visions of the Future"* <<http://www.ietf.org/proceedings/24.pdf>> »), que tous les participants à l'IETF connaissent par cœur : « *"We reject kings, presidents and voting. We believe in rough consensus and running code."* ». Les décisions à l'IETF ne peuvent pas être prises par un seul individu (roi ou président), mais elles ne doivent pas non plus être prises par un vote, pour les raisons expliquées plus loin. Au contraire, elles doivent s'appuyer sur le consensus approximatif, lui-même nourri d'expérience concrète (le « *"running code"* », par opposition à des organisations comme l'ISO qui ont développé des usines à gaz impossibles à programmer). La pratique doit toujours l'emporter : le code qui marche a raison.

Notez que le consensus est approximatif : avoir un consensus tout court (tout le monde est d'accord ou, au moins, se tait) serait l'idéal mais ce n'est pas réaliste. Demander un consensus complet serait donner un droit de veto à n'importe quel individu négatif qui s'oppose à tout (l'IETF, comme toute organisation humaine, a quelques trolls de ce calibre...).

Comme il n'y a pas de vote, l'IETF a développé une autre pratique inhabituelle, le mmmmmmm (*"humming"*). Lorsqu'on veut avoir une idée de l'opinion des gens, lors d'une réunion physique, on demande à chacun de se manifester par ce bruit produit en gardant la bouche fermée (ce qui assure un

certain secret, contrairement aux mains levées). En faisant des mmmmmmm plus ou moins forts, on peut même avoir un vote non-binaire. Ce rituel est souvent pratiqué pendant les réunions IETF.

Mais tous les participants à l'IETF ne comprennent pas forcément ces subtilités, surtout lorsqu'ils viennent d'arriver. « Pourquoi un mmmmm et pas un vote normal? » ou « Il y avait une majorité de mmmmm en faveur de la proposition B, pourquoi continue-t-on de discuter? » Même quand le groupe de travail est arrivé à un résultat, bien des participants ne comprennent pas comment il a été obtenu et pensent que le processus a produit un mauvais résultat. Paradoxe : il n'y a pas de consensus à l'IETF sur l'importance du consensus. Ou plutôt, il y en a un en parole (la phrase de Clark est très souvent citée) mais pas dans la réalité du travail.

Donc, premier point expliqué par ce RFC, en section 2, l'important n'est pas que tout le monde soit d'accord, mais qu'il n'y ait pas trop de désaccord. Imaginons un groupe qui doit trancher entre deux formats pour un protocole (mettons JSON et XML, pour prendre un exemple réel...). Il y a des gens pour et des gens contre chacune des deux propositions. Cela veut-il dire qu'il n'y aura jamais de consensus? Sauf que tout travail d'ingénierie (et la création d'une norme technique en est un) nécessite des compromis. Si on demande un accord de tous, on ne l'aura jamais (surtout avec le goût des ingénieurs pour les discussions sur la couleur du garage à vélos). En revanche, si on demande qu'il n'y ait pas d'objections bloquantes, on peut arriver à un résultat. Les partisans de XML ne vont pas tout à coup préférer JSON. Par contre, ils peuvent considérer que JSON, sans être leur solution préférée, est acceptable, qu'il n'a pas de défaut qui mettrait vraiment en péril le nouveau protocole. Il ne faut donc pas demander aux participants « êtes-vous heureux avec ce choix? » (il y aura forcément des râleurs) mais plutôt « ce choix a-t-il un défaut rédhibitoire (et, si vous répondez oui, argumentez)? » Bien sûr, il faut quand même qu'il y ait un certain nombre de gens qui soutiennent activement la proposition. Une solution technique qui ne rencontrerait aucune objection, mais pas non plus de soutien actif ne serait sans doute pas intéressante!

Autre point où il faut entrer dans les nuances, la différence entre compromis et compromission. Le mot « compromis » (en français, comme en anglais avec « *compromise* ») est utilisé dans deux sens différents. Le compromis est une bonne chose : en ingénierie, on a rarement des solutions idéales. Il faut en général arbitrer et, par exemple, décider de sacrifier du temps CPU pour consommer moins de ressources réseau ou le contraire, ou bien mettre de la redondance pour assurer l'intégrité alors qu'elle va diminuer les performances, ou encore préférer un format texte qui facilite le débogage à un format binaire plus efficace mais plus dur à traiter par les humains. Ce genre de compromis est un travail normal et essentiel de l'ingénieur, et l'IETF le fait.

En revanche, les compromissions sont néfastes : une compromission est quand on tient compte des opinions et pas des principes d'ingénierie. Par exemple qu'on accepte telle option, qui semble inutile, voire dangereuse, parce qu'une partie du groupe de travail trépigne et y tient absolument. On ajoute alors l'option pour obtenir leur accord (ou leur non-désaccord), faisant ainsi une compromission (ou un « *design by committee* », où le résultat n'enthousiasme personne, à force d'avoir intégré plein de demandes contradictoires). Ce genre de compromission (qu'il faudrait plutôt appeler capitulation) n'a pas réglé les problèmes techniques. Pire, on peut voir ce que le RFC désigne sous le nom de « *horse trading* » (comportement de maquignon?) « Je ne suis pas d'accord avec ta proposition sur l'algorithme de traitement de la congestion, tu n'es pas d'accord avec ma proposition sur le format des données, mettons-nous d'accord pour chacun soutenir la proposition de l'autre. » Cela permet d'arriver à un accord mais au prix de la qualité : les objections contre les deux propositions ont été ignorées, alors qu'elles étaient peut-être sérieuses techniquement et devaient être traitées comme telles. Ce comportement donne l'apparence du consensus alors qu'il est de la pure magouille.

Deuxième point important, en section 3, le fait qu'il faut traiter toutes les questions mais pas forcément les résoudre. Parfois, une objection technique va être résolue à la satisfaction de tous : celui qui l'a soulevée est convaincu que son objection était erronée, ou bien le groupe de travail a été convaincu qu'elle

était sérieuse et a modifié le protocole en cours de discussion. Mais, parfois, le groupe de travail ne change pas de direction et celui qui a soulevé l'objection est frustré. Est-ce un échec du consensus? Non, car il y a toujours « consensus approximatif ». Celui-ci est une solution de repli : l'idéal est le consensus complet mais, comme le dit le conte « Le meunier, son fils et l'âne », on ne peut pas toujours plaire à tout le monde et, de toute façon, il y a des gens qui ne sont jamais contents. La notion de consensus approximatif est faite pour ces cas-là : l'objection n'a pas été ignorée, elle a été sérieusement étudiée et le groupe de travail a délibérément décidé de continuer dans la voie originale. Compte-tenu de la forte personnalité de pas mal de participants à l'IETF, le consensus seulement approximatif, qui devrait être une exception à la règle du consensus, est souvent la norme.

Notez que ce consensus approximatif vu comme un repli lorsque le consensus idéal n'est pas atteint est une description nouvelle. Le RFC 1603<sup>1</sup> voyait plutôt le consensus approximatif comme exprimant le « point de vue dominant » du groupe de travail IETF. Cette vision quantitative (exprimée de manière encore plus directe dans son successeur, le RFC 2418) n'est pas partagée par ce nouveau RFC 7282, qui met l'accent sur le fait que les 0,1 % qui sont en désaccord ont peut-être raison (comme le disait un de mes enseignants à la fac, « la science n'est pas une démocratie »).

Et le mmmmmmm? La section 4 revient sur son rôle. D'abord, pourquoi ne vote-t-on pas, à l'IETF (certainement la première question que posent tous les nouveaux venus)? D'abord, l'IETF n'a pas de membres : il n'existe aucun processus formel d'adhésion. On s'inscrit à la liste de diffusion d'un groupe de travail et, hop, on est un participant à l'IETF. Impossible donc, de compter le corps électoral. Ensuite, le vote peut être une mauvaise chose : comme indiqué plus haut (même si l'expression n'est pas dans le RFC), la technique n'est pas démocratique. 51 % ou même 99 % d'un groupe de travail peut toujours voter X, c'est peut-être quand même Y qui est justifié techniquement. Le vote pourrait être un moyen de glisser sous le tapis les problèmes gênants, qui doivent au contraire être traités en pleine lumière. Une des raisons de l'utilisation du mmmmmmm est donc, précisément par le caractère folklorique de la procédure, de rappeler qu'on ne vote pas, à l'IETF (par comparaison avec, disons, un vote à main levée).

Le mmmmmmm sert donc à jauger la situation et à ordonner la discussion. Par exemple, s'il y a beaucoup plus de mmmmmmm objectant contre X que contre Y, partir des objections contre Y peut permettre d'optimiser le temps de discussion (sans pour autant considérer que Y a « gagné »).

Malgré cela, un effet pervers s'est développé, avec des gens considérant le mmmmmmm comme une sorte de vote anonyme. Ce n'est pas son but, le mmmmmmm devrait être utilisé au début (pour avoir une idée des points de vue en présence) pas à la fin pour clore le bec aux minoritaires. Le mmmmmmm n'est pas une méthode magique, et, s'il est mal utilisé, il a souvent les mêmes inconvénients que le vote classique (par exemple une extrême sensibilité à la rédaction des questions, ou même simplement à leur ordre).

Donc, le but, c'est le consensus? Pas du tout. La section 5 insiste sur le fait que le consensus est un moyen et pas une fin. Le consensus approximatif est un moyen d'obtenir la meilleure décision technique possible, en s'assurant qu'aucune objection, aucun problème, n'a été négligé. Au contraire, l'expérience des SDO qui votent a largement montré que cela mène à des mauvaises décisions, notamment suite à des compromissions pour obtenir une majorité, et suite au mépris des remarques parfaitement valides des minoritaires.

Alors, évidemment, la méthode IETF prend du temps et est souvent exaspérante, même pour les plus patients des participants. Et il ne faut donc pas se faire d'illusions : les magouilles politiciennes existent

---

1. Pour voir le RFC de numéro NNN, <https://www.ietf.org/rfc/rfcNNN.txt>, par exemple <https://www.ietf.org/rfc/rfc1603.txt>

aussi à l'IETF. Simplement, passer à un système plus rapide comme le vote permettrait d'avoir plus vite des normes de moins bonne qualité.

La section 6 revient sur la notion de majorité et sur l'approche quantitative, qu'appuyait le RFC 2418 (« *Note that 51% of the working group does not qualify as "rough consensus" and 99% is better than rough.* ») S'il y a 100 participants dans un groupe qui trouvent la proposition en cours excellente, et la soutiennent activement, et 5 participants qui sont en désaccord, prédisant, par exemple, « ce protocole va créer de la congestion et n'a aucun mécanisme de réaction en cas de congestion » et que personne ne répond de manière argumentée à ce point, alors, il n'y a pas de consensus, même pas de consensus approximatif. Une objection a été ignorée, pas traitée, et pas rejetée. Le faible nombre d'opposants n'y change rien, on ne peut pas parler d'un consensus, même pas approximatif. (Le RFC note que c'est pareil si la ou les personnes qui ont soulevé l'objection ne sont pas des participants réguliers, et qu'ils n'insistent pas après avoir fait leurs remarques ; les remarques en question doivent quand même être traitées.)

En sens inverse, et même si cela peut sembler anti-intuitif, on peut avoir 5 personnes qui soutiennent un projet et 100 qui sont contre, et qu'il y ait quand même un consensus en faveur du projet (section 7). Comment est-ce possible? Le RFC décrit une telle situation (et des cas analogues se sont effectivement produits) : un groupe de travail a peu de participants actifs. Ils sont arrivés à un consensus malgré les objections de l'un d'entre eux, objection qui a été étudiée et rejetée. Le « dissident » fait appel à une centaine de personnes, parfois des participants à d'autres groupes de travail mais souvent des employés de la même société que lui, qui ne connaissent rien au sujet mais font ce qu'on leur dit. Ils envoient tous un message sur la liste de diffusion, disant qu'ils soutiennent l'objection. Une bonne façon de traiter ces soutiens vides (sans contenu nouveau) est de leur demander « Nous avons déjà discuté de ce point et abouti à un consensus. Avez-vous des éléments techniques nouveaux qui justifieraient de rouvrir la discussion ? » En général, les commerciaux et marketeux qui avaient été recrutés pour faire masse dans la discussion ne répondent pas... Ils croyaient que l'IETF fonctionnait sur un système de vote et sont surpris de découvrir qu'on leur demande d'argumenter. Bien sûr, ce cas est un peu extrême mais la même méthode peut servir dans des cas moins dramatiques. Le principe du consensus sert à traiter une des plus grosses failles des systèmes de vote : le bourrage des urnes par des ignorants.