RFC 5218 : What Makes For a Successful Protocol?

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 25 juillet 2008

Date de publication du RFC : Juillet 2008

https://www.bortzmeyer.org/5218.html

Qu'est-ce qui fait qu'un protocole a du succès ou pas ? Pourquoi certains restent-ils dans une petite niche alors que d'autres accèdent à la domination mondiale ? Et y a t-il des leçons à tirer de ces succès et de ces échecs, pour améliorer les prochains protocoles ? C'est à ces questions que s'attaque le dernier RFC de l'IAB, « qu'est-ce qui fait un succès ? ».

Il n'y a évidemment pas de réponse simple et univoque. Le RFC 5218 [1] ne tente bien sûr pas de trouver une recette simple qui garantirait le succès. Il liste plutôt des facteurs qui jouent un rôle, essayant d'évaluer l'importance de ce rôle. La moitié du RFC est constituée d'études de cas où plusieurs protocoles sont étudiés du point de vue de ces facteurs.

Ce qui est sûr, c'est que la qualité technique du protocole n'est qu'un des facteurs, et pas le plus important.

Au sein des protocoles IETF, pourquoi Diameter (RFC 3588, puis RFC 6733) n'a t-il jamais remplacé Radius (RFC 2865) ? Pourquoi SNMP v1 (RFC 1157) continue t-il à être tant utilisé ? Et, bien sûr, même s'il est très peu cité dans le RFC, pourquoi IPv6 (RFC 2460) n'a t-il pas remplacé IPv4 (RFC 791) ? Pourtant, à chaque fois, le protocole présenté comme « le successeur » avait bénéficié d'un marketing vigoureux.

Entre protocoles IETF et protocoles d'autres origines, pourquoi IP a t-il écrasé IPX, qui était probablement techniquement supérieur, ou bien OSI, qui bénéficiait d'un soutien politicien et bureaucratique marqué ?

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc5218.txt

La section 1.1 du RFC commence par tenter de définir ce qui fait le succès. BGP (RFC 4271) est un succès, bien qu'il ne soit installé que sur un petit nombre de machines et qu'il ne soit utilisé directement que par une petite minorité de personnes. DHCP (RFC 2131) est un autre exemple d'un succès qui ne se voit pas forcément, puisqu'il n'est utilisé qu'à l'interieur d'un site (DHCP est un « protocole TCP/IP » mais pas vraiment un « protocole Internet »). Même chose pour OSPF (RFC 2328). En revanche, les protocoles célèbres, utilisés à travers l'Internet, sont des succès incontestables comme HTTP (RFC 2616), SIP (RFC 3261) ou le DNS (RFC 1034).

La section 1.2 essaie de quantifier cette question en parlant des **deux dimensions** du succès : les usages et l'échelle. Un protocole est conçu pour certains usages. Le succès signifie souvent qu'il est utilisé pour d'autres usages. Un protocole est prévu pour une certaine échelle d'utilisation et le succès va souvent le conduire à la dépasser largement (par exemple, il y a bien plus de machines connectées à l'Internet que n'en prévoyaient ses concepteurs, même dans leurs rêves les plus fous). Le RFC invente même le concept de « succès fou » ("*wild success*") pour désigner les protocoles qui, comme IPv4, ont dépassé toutes les espérances.

HTTP (section 1.2.1) est un exemple encore meilleur d'un succès fou. Il a dépassé les usages prévus (puisqu'on se sert de HTTP en dehors du Web, par exemple pour REST et XML-RPC, voire pour traverser les coupe-feux et « tunneler » le trafic). L'ampleur de son déploiement a aussi largement dépassé les ambitions initiales de Tim Berners-Lee !

Plus complexe, toujours en section 1.2.1, est le cas d'ARP (RFC 826). En ampleur du déploiement, c'est un grand succès. Mais ses usages ont été réduits : conçu pour fonctionner sur n'importe quel type de réseau local, il n'a jamais été déployé que sur Ethernet.

Alors, est-ce souhaitable pour un concepteur de protocole de voir son bébé devenir un « succès fou » ? Ce n'est pas évident, comme le note la section 1.3. Avoir un succès fou a des conséquences, certaines bonnes et d'autres mauvaises. Par exemple, le succès peut entraîner le protocole aux limites de ses capacités quantitatives, « problème » que connait actuellement IPv4, dont les quatre milliards d'adresses possibles ne sont plus suffisantes <http://www.potaroo.net/tools/ipv4/index.html>. (À l'époque de sa conception, le modèle dominant était « un ordinateur par organisation ». Il est passé ensuite à « un ordinateur par département » puis « un ordinateur par personne » et le modèle actuel de plusieurs ordinateurs par personne est une des causes de l'épuisement des adresses IPv4.)

De même, le succès peut aggraver ou révéler des problèmes de sécurité : le protocole qui réussit va attirer l'attention des craqueurs.

Et l'échec ? Symétrique du succès, c'est le sort de certains protocoles. La section 1.3 essaie de l'analyser. D'abord, il ne faut pas forcément être pressé. Au début, un protocole n'a aucun déploiement et aucune implémentation. Ce n'est qu'avec le temps qu'on peut dire que le protocole a réussi ou échoué. HTTP a mis plusieurs années à décoller, par exemple. IPv4 est resté le protocole d'un petit réseau inconnu des décideurs et des journalistes pendant de nombreuses années.

Les protocoles réseaux ont un problème particulier à surmonter, celui de l'œuf et la poule (un terme que critique notre RFC, d'ailleurs). En effet, le réseau ne sert à rien si on est tout seul. Si j'invente un protocole de courrier électronique supérieur à SMTP (avec l'expérience qu'on a aujourd'hui, ce n'est pas très difficile), ce protocole, si réussi soit-il, ne servira à rien puisque j'en serai le seul utilisateur et que je ne pourrai donc envoyer du courrier à personne. Le cercle vicieux s'enclenche facilement : personne n'a déployé le nouveau protocole donc les auteurs de logiciels ne se pressent pas pour l'intégrer dans leurs programmes donc les utilisateurs ne s'en servent pas, donc il n'y a pas de déploiement, etc. Tout le monde aurait intérêt à ce que le nouveau protocole réussisse mais les premiers convertis supporteraient

tous les coûts. En l'absence d'une autorité centrale qui pourrait ordonner le déploiement du nouveau protocole, bien des propositions de réforme ont ainsi été victimes de ce que les économistes appellent un échec du marché <https://www.bortzmeyer.org/ipv6-et-l-echec-du-marche.html>.

Quelles sont les méthodes pour faire face au problème de l'œuf et de la poule ? La section 1.3 en cite plusieurs :

— Résoudre un problème immédiat et brûlant, ce qu'a fait, par exemple, le NAT (RFC 3022).
— Fournir un « service qui tue » (*"killer app"*), comme l'ont fait les services de distribution de fichiers en pair-à-pair, service tellement utile que les utilisateurs sont prêts à faire des efforts, par exemple à reconfigurer leur routeur.
— Être invisible, c'est-à-dire ne nécessiter aucun changement.
— Réduire les ambitions, en diminuant les usages prévus, ce qui rend le succès plus facile. C'est ainsi que la diffusion restreinte (*"multicast"*) sur tout l'Internet, telle qu'envisagée à l'origine, a été un échec mais cette même diffusion restreinte est largement pratiquée sur les réseaux locaux.
— Obtenir une aide ou une directive gouvernementale. La question est chaudement disputée, les adorateurs du marché aimant prétendre que l'Internet s'est créé tout seul, par le seul jeu non régulé des forces du marché. Rien n'est plus faux, l'Internet et son prédécesseur Arpanet ont vécu pendant quinze ans aux crochets de l'état états-unien. Mais l'intervention de l'État (à part sous une dictature extrême du genre Ivan le Terrible) ne marche pas toujours, les acteurs gardant leur liberté. Le RFC cite l'intervention du gouvernement japonais en faveur d'IPv6 mais, si cette aide a permis à Kame de faire un travail très intéressant sur IPv6, ce protocole reste minoritaire au Japon comme ailleurs. De même, les protocoles OSI ont nettement perdu face à TCP/IP malgré un soutien gouvernemental très obtus (par exemple en France). Et, dans un domaine différent de celui des réseaux, le langage de programmation Ada ne s'est pas imposé, en dépit d'une consigne officielle de n'utiliser que lui pour tous les programmes de l'armée états-unienne (le plus gros consommateur de logiciels du monde).

La section 2 explore les différentes causes de succès, en précisant bien qu'aucun succès n'a réuni **toutes** ces causes. 2.1 se spécialise dans les raisons du succès « de base », 2.2 dans celles du succès fou.

D'abord, le protocole doit évidemment avoir un intérêt et cet intérêt doit compenser les coûts (section 2.1.1). Si les coûts liés au matériel sont les plus visibles, ils ne sont pas forcément les plus importants. Ainsi, tout protocole qui nécessite de re-former les utilisateurs a un travail plus difficile pour s'imposer car le coût de cette formation est non-trivial. Les coûts liés à l'invalidation des anciens calculs économiques sont également à prendre en compte : si un FAI fonde son modèle économique sur des connexions Internet intermittentes, un protocole qui permet d'être connecté tout le temps comme l'ADSL, va remettre en cause ce modèle (de façon significative, mais étonnante pour un RFC, le paragraphe sur l'économie est le plus long de la section).

Et quels sont les bénéfices possibles ? Résoudre un problème lancinant (DHCP a ainsi mis fin au cauchemar des administrateurs réseaux courant de machine en machine pour faire un changement), permettre des choses qu'on ne faisait pas avant, ou rendre simplement plus efficace les choses qu'on faisait déjà (de tels protocoles sont souvent plus simple à déployer car ils ne remettent pas en cause les usages). Si le coût initial est élevé, le protocole peut avoir du mal à s'imposer, même si les bénéfices futurs sont prometteurs.

En outre, coûts et bénéfices ne sont pas équitablement répartis. Les techniques de NAT sont très coûteuses pour les développeurs d'applications, qui doivent coder des contournements compliqués mais apportent des bénéfices aux FAI, qui font des économies d'adresses IP. Tout dépend donc de qui décide. Le RFC note donc que le succès d'un protocole vient souvent de « l'alignement des coûts et des bénéfices », c'est-à-dire du fait que les bénéfices viennent à ceux qui supportent les coûts initiaux (même si, comme dans le cas du NAT, les coûts finaux sont payés par d'autres).

Ensuite, pour être un succès, le protocole a tout intérêt à être déployable petit à petit (section 2.1.2). En effet, le dernier "*flag day*" (tout l'Internet change de protocole le jour J à l'heure H) a eu lieu en janvier 1983 lors du déploiement d'IPv4 (et du passage à TCP/IP). Aujourd'hui, il est complètement impossible de changer tout l'Internet d'un coup et le nouveau protocole doit donc pouvoir s'intégrer à l'Internet existant (le grand drame d'IPv6 vient de là).

D'autres causes de succès non technique sont la disponibilité du code en logiciel libre ou quasi-libre (sections 2.1.3 et 2.1.4), le fait que le protocole soit ouvert <https://www.bortzmeyer.org/formats-ouverts.html> (section 2.1.5) et maintenu par une SDO selon un processus ouvert (section 2.1.6, qui est un peu un plaidoyer "*pro domo*" pour l'IETF). Le succès d'IPv4 contre IPX ou OSI (ce dernier cas n'est pas cité par le RFC mais est encore plus net) tient largement à ces points. TCP/IP a gagné en bonne partie par sa disponibilité dans les Unix de Berkeley.

Le fait que la norme soit « ouverte » joue un rôle important, ce qui explique aussi la vigueur des débats dans le groupe de travail IPR <http://tools.ietf.org/wg/ipr> de l'IETF. Les RFC ne sont pas les normes les plus ouvertes du monde, par exemple leur licence (RFC 5378) limite les usages qui peuvent en être faits...

Et la technique, alors, elle ne joue aucun rôle ? Un peu quand même, argumente la section 2.1.7, qui explique qu'un protocole bien conçu a davantage de chances.

Mais le succès est une chose et le succès fou en est une autre. Qu'est-ce qui cause un succès fou ? L'extensibilité et l'absence de limites aident (sections 2.2.1 et 2.2.2). DECnet était limité par construction à 65 536 machines, ce qui garantissait son échec à terme, même s'il avait connu de bons débuts.

L'un des points où il y a la plus grosse différence entre le succès et le succès fou concerne la sécurité (section 2.2.3). Au début, il n'est pas nécessaire qu'un protocole soit sûr (ce point est développé également en section 3). Compte-tenu des divers coûts associés à la sécurité, notamment les difficultés d'usage, faire l'impasse sur la sécurité est en général un bon moyen de réussir au début. Mais, lorsque le protocole devient très répandu, les craqueurs se précipitent dessus et cherchent à le subvertir. C'est alors que les impasses qui avaient été faites sur la sécurité peuvent se payer cher. Néanmoins, foncer d'abord et réfléchir ensuite à la sécurité semble être une bonne stratégie (les protocoles qui intègrent la sécurité dès le début ne sont souvent jamais déployés et donc jamais attaqués).

Notons que l'annexe A, qui représente la moitié du RFC, décline ces analyses pour une série d'étude de cas. C'est ainsi que sont examinés :

— Protocoles Web (HTTP et HTML) contre Gopher (section A.1). La qualité technique n'a guère joué (HTML à cette époque était très pauvre et, contrairement à ce que dit le RFC, n'avait même pas les formulaires). Mais le déploiement était simple et apportait des bénéfices immédiats.
— IPv4 contre IPX (section A.2). Il existait d'innombrables protocoles réseaux à l'époque. Tout administrateur réseau devait en gérer plusieurs. Aucun des protocoles concurrents n'était déployable progressivement mais la base installée était faible. Certains de ces protocoles avaient des limites fondamentales (comme AppleTalk) qui leur interdisaient d'être le protocole d'un réseau mondial. IPX n'avait pas ces défauts et avait bien des avantages techniques sur IPv4 (comme l'autoconfiguration, alors que DHCP n'existait pas encore). Mais la grande force d'IPv4 était la disponibilité d'une mise en œuvre libre, dans BSD (financée par des fonds publics). Même chose pour la disponibilité d'une norme.

— SSH (section A.3). Ce protocole (RFC 4251) a remplacé telnet et rlogin en très peu de temps. Très simple à installer et à utiliser (beaucoup l'ont adopté uniquement pour la redirection des sessions X11, un énorme progrès à l'époque), ne nécessitant pas d'infrastructure centrale (contrairement aux solutions à base de Kerberos), SSH, malgré un statut légal peu clair au début, s'est vite imposé. Les puristes de la sécurité avaient pourtant critiqué cette simplicité (bien que beaucoup de solutions de sécurité aient échoué en raison de la grande complexité qu'elles imposaient aux utilisateurs). Le modèle TOFU ("*Trust On First Use*") de SSH leur semblait une hérésie. Mais l'alternative était du telnet avec zéro sécurité. (Aujourd'hui, des techniques comme Perspectives <https://www.bortzmeyer.org/perspectives-ssh.html> tentent d'améliorer le système TOFU.)
— Diffusion restreinte ("*multicast*") entre domaines distincts (section A.4). C'est cette fois un échec qui est analysé. Le "*multicast*" (RFC 5110) a souvent été présenté comme indispensable aux services multimédia sur Internet mais la réalité est qu'il n'a jamais connu d'utilisation significative. Pourquoi ? Il manquait de la plupart des facteurs de succès. Il n'est pas évident qu'il répondait à un problème réel (la diffusion est utile si tout le monde regarde la même chaîne de télévision en même temps, comme dans les années 60, mais elle ne sert à rien pour la VoD). Il nécessite des grands changements dans tous les routeurs du chemin. Bref, les coûts étaient importants et les bénéfices peu clairs.
— WAP (section A.5) est un autre échec, qui avait pourtant bénéficié d'un marketing très lourd et peu subtil, composé essentiellement d'affirmations tonitruantes comme quoi « 90 % des accès au Web se feraient en WAP dans cinq ans » et autre fumage de moquette de consultant. Coincé entre le classique HTTP et le plus perfectionné I-Mode, WAP n'apportait rien de précis. Très fermé (licences à acheter, normes non accessibles, évolution pilotée par une organisation où le coût d'adhésion était de 27 000 $), WAP n'est pas allé loin. Il est amusant a posteriori de compter le nombre d'articles sur WAP qui étaient parus dans la presse pour décideurs, alors qu'il n'avait connu aucun déploiement, par rapport au nombre d'articles qu'avait eu IP alors que l'Internet était déjà massivement utilisé.
— La lutte du protocole IETF Radius (RFC 2865) contre le protocole de Cisco Tacacs (section A.7) est un autre cas où la disponibilité d'une norme ouverte et sans entraves a été le facteur dominant. Tacacs a évolué par la suite, une implémentation libre est apparue, ainsi qu'une norme (RFC 1492) mais c'était trop tard.
— Plus douloureux est le succès du NAT (section A.8). Apportant un bénéfice immédiat à celui qui le déploie (le "*IP masquerading*" de Linux, première mise en œuvre répandue du NAT, avait été un succès foudroyant et avait beaucoup contribué au décollage de Linux), le NAT, propulsé par la pénurie d'adresses IPv4, s'est répandu partout, en dépit de l'opposition vigoureuse de l'IETF. Le fait qu'il apportait un bénéfice à court terme et qu'il était déployable sur un site sans aucune coordination avec les autres lui ont permis de progresser très vite.

La section 3 sert de conclusion au RFC. Elle estime que les facteurs de succès les plus importants sont le fait que le protocole apporte un avantage réel et le fait qu'il puisse être déployé progressivement, sans "*flag day*".

Les qualités techniques du protocole sont secondaires et le RFC note, à juste titre, que beaucoup de protocoles ayant connu un succès fou ne passeraient pas l'examen de l'IESG aujourd'hui...

La conclusion finale est donc que l'examen par l'IETF des nouveaux protocoles devrait donc inclure l'étude des facteurs de succès.

Enfin, je note que l'IAB n'a guère mentionné des cas où l'échec est plus gênant pour l'IETF comme le peu de déploiement d'IPv6...