Economics of Information Networks

Stephen Turnbull

Division of Policy and Planning Sciences

Lecture 4: December 10, 2020

Abstract

We continue discussion of the "modern" economics of networks, which considers the effect of structure of networks on economic (including social) behavior.

The structure of the Internet

- Why is it the *Inter*net? Why is it *the* Internet?

  These are the big questions about the structure of the Internet, which is no longer just a computer-to-computer network. The Internet, the telephone system, and cable television networks have merged (and created new infrastructure such as "the cloud" as well).

- The second we have discussed already. Networks create network externalities. Even if not as dramatic as Metcalfe's Law, each node in the network gets greater benefits as the size of the network (number of nodes) increases.
- There are very large economic benefits to network growth, or to merging networks that were originally separate, even if they seem to be of different "kinds" (such a data networks, voice networks and video networks).

The structure of the Internet, *cont.*

- It's the *Inter*net because it is an *internet*: a network composed by connecting networks together.
- These networks
  1. use different *physical* media (wireless, fiberoptic, metal wires)
  2. use different *protocols* (LTE, 5G, Ethernet, token ring, PPP)
  3. are maintained by different *organizations* (telephone companies, railroads, power companies, universities, corporations, even you.
- This is all very complex. Ask me about *email*, for example (I have about 35 separate standards for email bookmarked!) It isn't really possible to discuss the network structure of the Internet without the details. So for detail, we'll consider a much more uniform network which is more purely an *information* network: the World Wide Web.

What is the World Wide Web?

- The World Wide Web (WWW, or just "the web" for short) is an information network supported by the Internet.
- It is enabled by four central *protocols*, which (on the Internet) is a fancy way of saying "formats for information." The Internet handles the details of moving information around, while the WWW focuses on what information we want to convey.

  1. The *Domain Name Service* (DNS) allows us to access other systems on the Internet by name.
  2. The *Universal Resource Locator* (URL) system gives names to resources we want to access.
  3. The *HyperText Transport Protocol* uses the Internet to move information around as packets of bits.
  4. The *HyperText Markup Language* (HTML) provides the ability to link resources together.

The Domain Name Service

- The Internet Protocol (IP) addresses used by Internet hosts (both servers and clients) are either 32 bits (4 bytes, same as most emoji) in version 4, or 128 bits in version 6.
- These addresses have a numerical structure that made it easy to direct information to the right place—easy for a computer, that is. I used to remember several IPv4 addresses because they were local and I had to teach them to my computers, but 128 bit I wouldn't. Even in my generation few people remembered IPv4 addresses.
- The *Domain Name Service* (DNS) allows us to access other systems on the Internet by name. It translates names humans can remember to IP addresses that allow for fast routing of packets.
- It is implemented by a system of *nameservers* maintained by Internet providers.

Universal Resource Indentifiers

- The *Universal Resource Identifier* (URI) system gives a flexible consistent way to name the resource we want.
- One kind of URI is the Universal Resource Name (URN). You are probably familiar with the DOI (*digital object identifier*) scheme used to identify journal papers as one example. The point is that you don't need to know where the paper is. Each URN scheme makes up its own rules about locating resources.
  - URNs are often used for *distributed archives* or even *peer-to-peer* systems such as Usenet (netnews).
- You're probably more familiar with the *Universal Resource Locator* (URL). A URL has a standardized structure, composed of a scheme, an authority, a path, a query, and a fragment.

Universal Resource Locators

- It lookes like this:

  `scheme://authority/path1/path2?key1=value1;key2=value2#fragment`

- The *scheme* says *how* to use the Internet to fetch the resource.
- The *authority* (usually a DNS *domain* naming a server) decides if access is allowed, and if so, interprets the path, query, and fragment.
- The *path* tells the server where to get the resource. The format is modeled on a file system consisting of a hierarchy of folders.
- The *query* ("key1=value1;key2=value2") directs dynamic features such as database lookups.
- The *fragment* identifies a portion of the whole resource. It's typically used to jump to a position on a page (including internally, for end notes and bibliographies).

The HyperText Transport Protocol

- The *HyperText Transport Protocol* (HTTP) is implemented by every webserver and client. It uses the Internet to move information around as packets of bits. It's very flexible.
- It's *bidirectional*: with permission of the authority, able to upload and download resources.
- It can be used to *stream* resources as they become available, and to conduct "conversations."
- There used to be quite a few protocols used on the Internet to upload and download resources, such as the *File Transfer Protocol* (FTP), Gopher, remote copy (rcp), even email, as well as more specialized protocols such as DNS, and the *Network News Transport Protocol* (NNTP). Recently more and more of these specialized systems have been converted to use HTTP to transfer their data.

The HyperText Markup Language

- The *HyperText Markup Language* (HTML) provides the ability to link resources together.
  - It also provides many features for formatting and presentation, especially in combination with *cascading style sheets* (CSS). These features are not relevant to our concern with information networks.
- Linking is in the name *hypertext*, coined by Ted Nelson to mean a linked document system in 1963.
- In HTML, links are asymmetric (arrows).
  - The WWW is a *directed multigraph*.
  - In Nelson's hypothetical hypertext system *Xanadu*, links were automatically symmetric.
  - In HTML symmetric links can be emulated with a link in each direction, and these are frequently used for footnotes and bibliographic citations.

What is "Web 2.0"?

- "Web 2.0" isn't really a change in the web itself. The Internet itself and the WWW protocols have evolved, but to content creators and users alike, it doesn't look much different.
- The original WWW was "static", because both hardware and browsers were slow.
  - Slow hardware meant that content on *servers* was *static*: stored in files.
  - Slow browsers meant that content as *displayed* was *static*: once a page was displayed, to make it change you needed to fetch a new page (or a new version of the page).

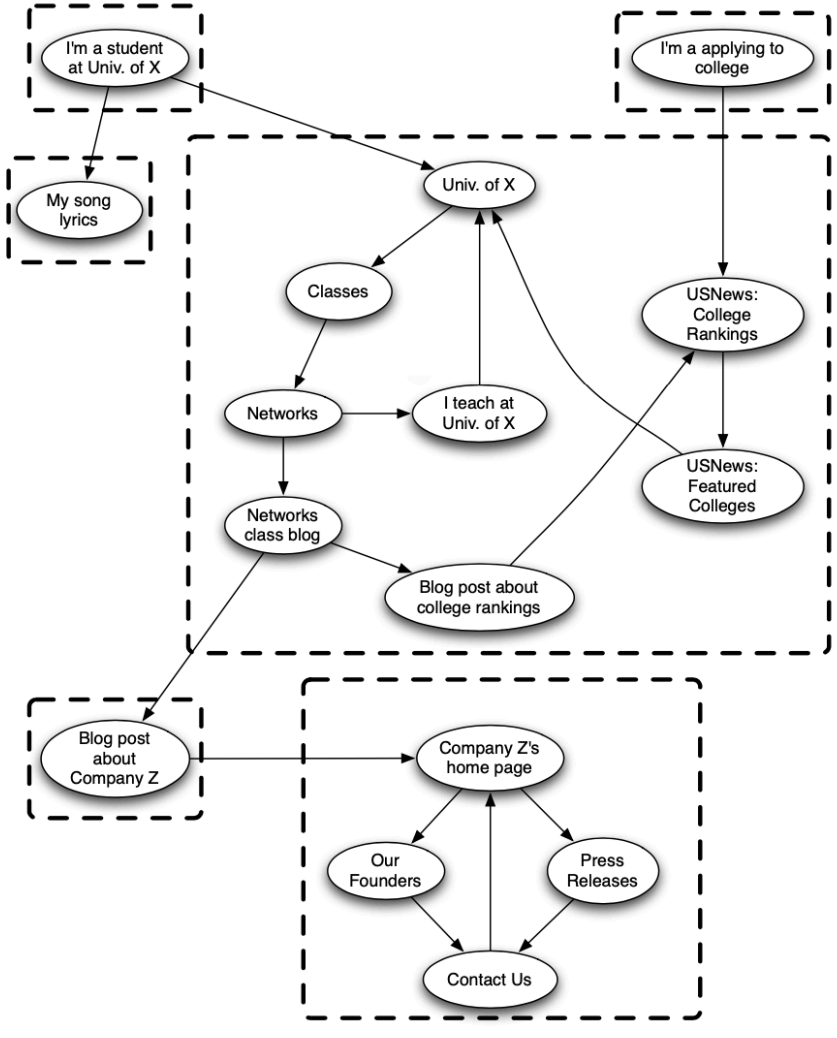
Web 2.0 is dynamic

- The dramatic difference is that "Web 2.0" is *dynamic*.
  - Servers are very likely to generate content by looking it up in a database or computing it by a program on every access, then formatting it for transmission as a web document.
  - These web documents can contain Javascript *code* that is executed by the browser to change the display, rather than fetching new content from the server.
- But the process of fetching documents, and the formatting of the documents, is the same as ever.

Fine structure of the WWW

- The WWW is a *directed multigraph* of *resources.*

There can be multiple links to the same resource from one resource. Having traversed a link, you can't go back without an external memo (such as browser history) or a reverse link (but is there are several, you can't know which is "back."



About resources and links

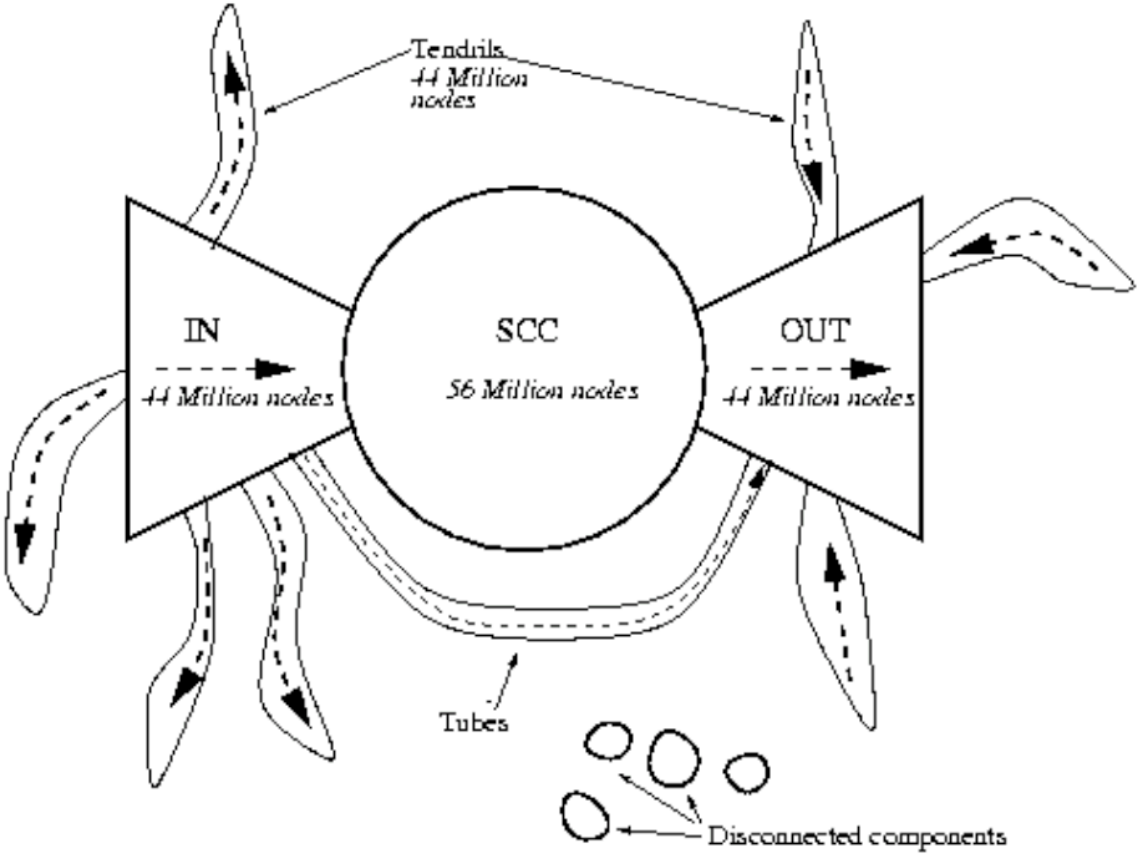
- In the WWW, a *resource* is anything that can be addressed by a URL. It doesn't even need to return anything for the client to display. Examples:
  - Text pages (HTML, plain)
  - Presentation resources (stylesheets, Javascript)
  - Multimedia (images, audio, video)
  - Arbitrary files (OA files, programs)
  - Form submissions (credentials)
  - Database lookups
- *Links* are indicated by URLs. There are two types:
  - User-initiated (`A` or anchor links)
  - Automatic (`IMG` or image links)
  - The distinction is browser-dependent

The global structure of the World Wide Web

- Although it is not a *social networking platform* as we understand that term now (*e.g.*, Facebook, Twitter), it is a platform for creating social networks.
- Linking to another page is a social activity. One can use it as a memo to oneself (that's what your browser's bookmark window is), but generally it's a service for your readers.
  - At present, just *readers*. But we already have voice recognition with Siri, Echo, and so on, and you can imagine virtual reality systems in which you can activate links by some kind of gesture, or even via eye motion.
- The Web truly is global. Some countries (Russia, China, Kazakhstan are well-known) impose substantial restrictions on access to international resources, but even they allow most connections.

Overall structure of the WWW

- The WWW is "almost" connected. It has giant component that contains most of the resources, and all of the links between them.
- The giant component has a sort of "fuzzy bowtie" structure.



The "bowtie" structure

- This diagram is based on the state of the Web in about 2000, using the index of the AltaVista search engine operated by the Digital Equipment Corporation (DEC).
- The central "knot" of the "bowtie" is a giant *strongly connected* component (SCC, defined as a maximal subset of nodes all of which are reachable by a directed path from any other node in the SCC), containing about 1/4 of the resources (or pages).
- The **In** set of the bowtie is the set of nodes from which the SCC is reachable.
- The definition of the **Out** set is an exercise.
- A *tendril* is a set of related paths not part of the SCC, which either
  1. contain nodes all of which can reach the **Out** set, or
  2. contain nodes all of which are reachable from the **In** set.

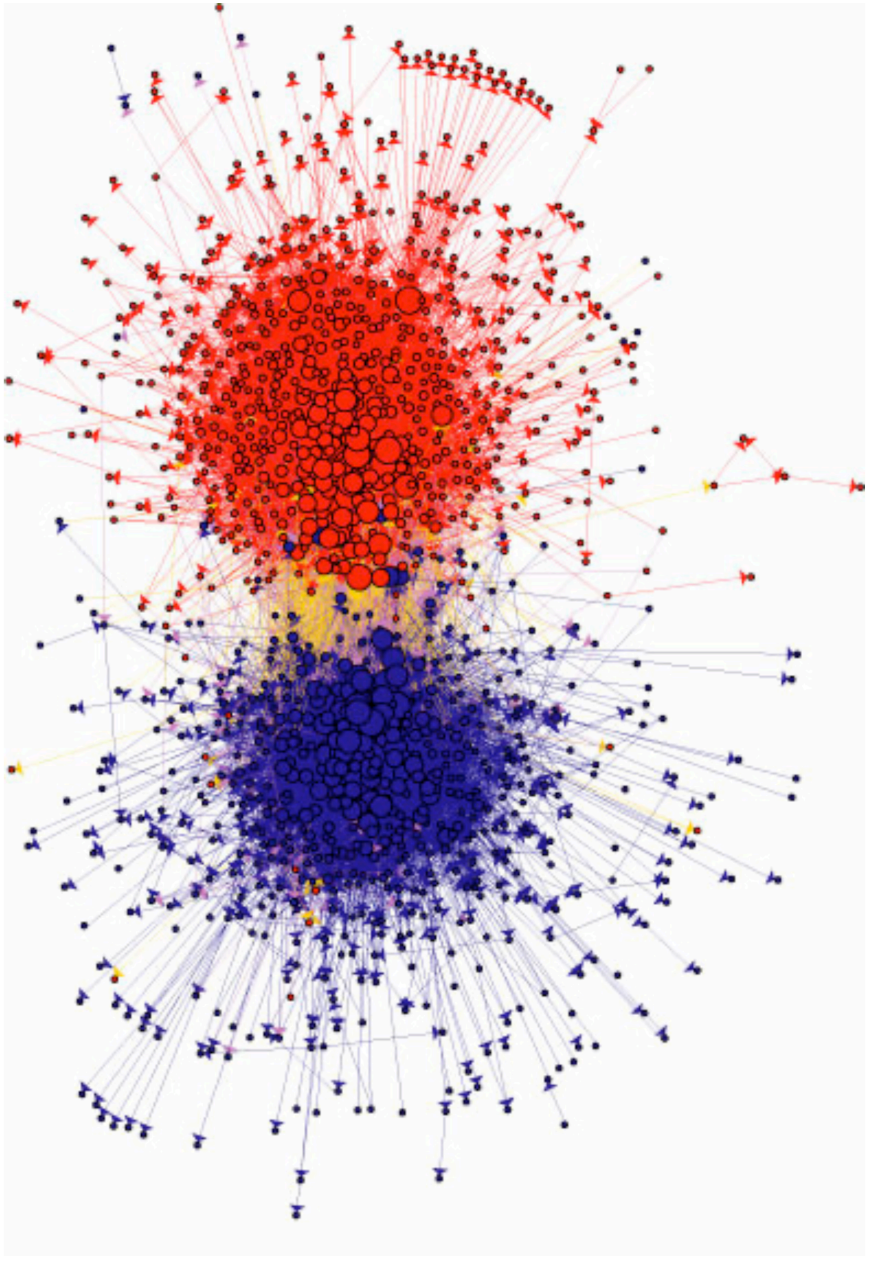
Homework: Due 17 December 2020

This task will be given a number and proper instructions later.

1. Why do you think the *disconnected components* at the bottom don't have an estimated number of resources?
2. Give the definition of the "out" side of the bow.
3. How are *tubes* related to *tendrils*?
4. Why can't a tube run from the **Out** set to the **In** set?
5. (Requires knowledge of Internet servers) How do you think they got information about the **In** set?

Web subsets: Political blogs

Politics is often described as a spectrum from left to right, but the U.S. Constitution leads to a two-party system, and this seems to be reflected in political blogs.



Homework: Due 17 December 2020

This task will be given a number and proper instructions later.

1. The graph of the political blog space shows **Out** fringes for both red (progressive) and blue (conservative) "components," but no **In** fringe. Why do you think that is? (Hint: It's probably related to data collection method.)
2. Are there any tendrils in this graph? If "yes," describe where they are. If "no," explain why not?
3. What can you say about the depth of the **Out** fringes? Can you explain why this might be?

Page Rank and other mysteries

- It's very common for a Google search to claim millions of matches for a search. How do they decide which results to return first?
- It's complex, but one factor is the "importance" of the resource. The measurement of importance is still a hot topic of research, both in academia and as a trade secret of many companies. It is closely related to many other concepts of importance such as the impact of an academic publication.
- The obvious algorithm is described as *voting by in-link* in Kleinberg and Easley. The idea is simple: take all the pages that contain your search key, and rank them according to the number of times they're linked from other pages.
- The obvious algorithm has a major flaw. Some pages get many links, regardless of their content.
- There are two subtly different approaches: *hub and authority* ranking, and *Page Rank*.

Hub and authorities

- This approach divides pages into two kinds.
- *Authorities* are the pages that are likely to give you the information you want.
- *Hubs* are pages that have many links to possible authorities.

Hub and authority algorithm

The algorithm is *iterative*, with two *updating rules* and a *stopping condition*.

1. *Initialization.* Give each page a *hub score* of 1, and an *authority score* of 1.
2. Apply the *authority update rule*: Each page's new *authority* score is the normalized sum of the *hub* scores of pages that link to it.
3. Apply the *hub update rule*: Each page's new *hub* score is the normalized sum of the *authority* scores of pages that it links to.
4. If the stopping condition is satisfied, stop, and rank authorities by their authority score.
5. Otherwise, repeat from Step 2.

Homework: Due 17 December 2020

This task will be given a number and proper instructions later.

1. Why do the update rules have "normalized" in them?
2. Give a formula for normalizing scores. It must preserve ranking according to score.
3. Give as many reasonable stopping rules as you can think of. (I can immediately think of one exact stopping rule and two kinds of approximate rule.)

Page Rank

- The hub and authority algorithm makes especially good sense when ranking is extremely competitive, for example in *recommender* algorithms for products by competing vendors. The product home pages aren't going to link to each other! So the asymmetry of hubs and authorities in the algorithm matches social behavior.
- But in many other situations, a node will be considered "important" if it's endorsed by other "important" nodes.
- *Page Rank* is an algorithm invented by Google's founders and commonly used as the basis for many commercial and academic endorsement systems, based on this more symmetric idea.
- Page Rank is like a fluid that circulates through the network. It accumulates at the most endorsed pages.

Basic Page Rank algorithm

The algorithm is *iterative*, with one *updating rule* and a *stopping condition*.

1. *Initialization*. Give each of n pages a *Page Rank* of $1/n$.
2. Apply the *basic Page Rank update rule*: Each page divides its Page Rank equally among the pages it points to (keeping it all if it has no out-links), and updates its Page Rank to the sum of Page Rank it receives.
3. If the stopping condition is satisfied, stop, and rank pages by their Page Rank.
4. Otherwise, repeat from Step 2.

This algorithm has a major flaw: if there are strongly connected components with no out-links, they will collect all of the Page Rank. This would be very bad. *E.g.*, for academic papers, this means that student essays with no bibliography would accumulate all the Page Rank!

Revised Page Rank algorithm

The algorithm is *iterative*, with one *updating rule* and a *stopping condition*.

1. *Initialization.* Give each of n pages a *Page Rank* of $1/n$, and choose a number $s \in (0, 1)$.
2. Apply the *revised Page Rank update rule*: Each page takes s times its Page Rank and divides its Page Rank equally among the pages it points to (keeping it all if it has no out-links), takes the remaining $1 - s$ of it Page Rank and divides it equally among *all* pages, and finally updates its Page Rank to the sum of Page Rank it receives.
3. If the stopping condition is satisfied, stop, and rank pages by their Page Rank.
4. Otherwise, repeat from Step 2.

This algorithm gives a different Page Rank vector for each value of s.

Homework: Due 17 December 2020

This task will be given a number and proper instructions later.

1. Why doesn't either Page Rank update rule need normalization?
2. Are the stopping conditions you designed for the hub and authority algorithm suitable for the Page Rank algorithms? Explain.
3. Do you think the "circulating fluid" analogy to Page Rank is a good one? Explain.