

## Class Notes

### **Class 1 – Overview, Course Rules**

#### Overview

- Topic: using computer and network technology to help run businesses and other organizations
- Won't focus especially on "managers"
- Will combine "Top-down" descriptive learning (the RTP book) with "bottom-up" learning by example (Microsoft Access and GB book)

Rules and Procedures – see the syllabus and schedule

### **Class 1 – Basic Definitions and Concepts (see chapters 1,2 of RT)**

#### Data, Information, and Knowledge:

- *Datum* is singular, *data* is plural (book says "data item" and "data items"). A datum is a "particle" of information like "12" or "Q".
- *Information* is data structured and organized to be useful in making a decision or performing some task.
- *Knowledge* denotes "understanding" of information:
  - Example from (old edition of) book: company analyzes its recruiting data and concludes that recruits from school X tend to have good outcomes only if their GPA's are at least 3.0. In future, based on this "knowledge", they screen applicants from school X by their GPA's, only interviewing those with at least a 3.0 GPA.
  - One common kind of knowledge representation in computers is called "artificial intelligence" (AI). It got a lot of hype in the 1980's, and then went somewhat out of fashion, but it is still growing gradually. We will not discuss it much, and stick to "information" instead.

#### *Information systems* (definition of some basic terms)

- The ways that organizations
  - Store
  - Move
  - Organize
  - Manipulate/processtheir information
- Components that implement information systems – in other words, *Information Technology*

- Hardware – physical tools: computer and network hardware, but also low-tech things like pens and paper
- Software – (changeable) instructions for the hardware
- People
- Procedures – instructions for the people
- Data/databases
- Information systems existed before computers and networks – they just used relatively simple hardware that usually didn’t need software (at least as we know it today). Strictly speaking, this course is about “CBIS” (Computer Based Information Systems). Because of the present ubiquity of such systems, we usually leave the “CB” to be implicit.
- Impact of computer and network hardware and related software/services (RT Table 2.1):
  - Can perform numerical computations and other data processing much more quickly, accurately, and cheaply than people
  - Can communicate very quickly and accurately
  - Can store large amounts of information quickly and cheaply; retrieval can often be very rapid
  - Can automate tasks and processes that previously required human labor (various degrees possible, of course)
  - Information doesn’t have to be “stuck” with particular things, locations, or people
  - Can have a downside
    - Small glitches can have much wider impact (minor software bug grounds all aircraft in Japan)
    - Fewer people in the organization understand exactly how information is processed
    - Sometimes malfunctions may go unnoticed (American Airlines yield management story)
- *Information architecture* is the particular way an organization has arranged its information systems: for example, a particular network of computers running particular software supports the marketing organization, while another network of computers running different software supports the production facilities, etc.
- *Information infrastructure* consists of the hardware and software that support the information architecture, plus the personnel and services dedicated primarily to maintaining and developing that hardware and software.
- *Application* and *application program* are somewhat fuzzy terms, but typically denote computer software and databases supporting a particular task or group of tasks.
  - Example from book: HR uses one application to screen job applicants and another to monitor employee turnover
  - A classic business IT problem: applications that don’t communicate with one another (effectively)

## ***Class 1 – Types of information systems (RT 2.1)***

Refer to RT Figure 2.2:

- *Departmental information systems*, or *functional area information systems* are designed to be operated within a single traditional functional department of an organization such as

sales, human resources, or accounting. In the early days of CBIS, these were often the only kind of systems that were practical.

- *ERP (Enterprise Resource Planning) systems* are a relatively extreme reaction to the problem of poorly integrated functional area systems, offered by vendors such as SAP and Oracle. They aim to support the entire organization's needs with essentially a single integrated system. They have enormous potential benefits, but are also notoriously tricky and expensive to configure and install.
- *Transaction Processing Systems (TPS)* gather data about everyday business events in "real time" as they occur. Examples:
  - You buy 3 items at a local store
  - A shipment of coffee beans arrives at a local distribution center
  - A package is unloaded from a FedEx or UPS aircraft

All of these events are examples of *transactions* that may be immediately tracked by a TPS. Often, technology like barcodes and scanners makes tracking such transactions quicker, cheaper, and more detailed than it would otherwise be.

- Some other common terms we will define in more detail later in the course:
  - MIS – "Management Information System"
  - DSS – "Decision Support Systems"
  - ES – "Expert Systems"
  - EIS – "Executive Information Systems"
- An *Interorganizational System (IOS)* connects two organizations – for example, it may allow a company to automatically share inventory and backlog data with suppliers or customers.
- *Electronic Commerce* or *E-Commerce* refers to sales transactions in which at least one side of the transaction (buyer or seller), and perhaps both, is performed by a CBIS without direct human help.

## ***Class 2 – Role of the Information Systems Department (RT 2.4)***

- Setting up and maintaining modern computer and network hardware software requires specialized skills and knowledge, at least for firms beyond a certain size.
- This means that the organization needs a sub-organization responsible for IT support: the Information Systems Department (ISD). Names vary from organization to organization; for example, at Rutgers it's called RUCS.
- In the early days of CBIS, ISD's "owned" *all* the information infrastructure because other departments had very little understanding and even routine use of the systems required specialized skills.
- As computers became more pervasive and user-friendly, managing IT resources has become a cooperative venture between ISD's, the departments/functional areas, and "end users" (individuals). Drawing the lines of who is responsible for what can be tricky.
  - At Rutgers, for example, RUCS operates the central network infrastructure, certain key systems like Eden and the outgoing mail server, and works in a consulting/contracting role to support other sub-organizations.
  - Individual or departmental management of resources will tend to be more responsive and understand user/departmental needs better

- Central management will tend to have a better understanding of the technology in general, may promote better integration/coordination between departments, and can lead to economies of scale
- Some balance needs to be struck
- This is a generic management issue that applies to lots of areas besides IT. For example, should each product division have its own product development engineering department, or should multiple divisions share an engineering department?
- In many organizations, the ISD has evolved (or should evolve) into a “business partner” with other departments, and not just a support organization – see below.

## ***Class 2 – Global Internet Environment (RT Chapter 1)***

While improvements in computer processing speed and storage capacity have been important, the most critical ingredient in the “global, web-based environment” has been *networking* – the interconnection of multiple computers, and specifically the internet.

A brief history of internet technology (we’ll do a bit more later in the course):

- The key to the present internet is the “TCP/IP” network technology developed in the 1970’s
- TCP/IP was in wide use by research/academic US computer users by the mid 1980’s. Typical applications were remote computer use (TELNET), file transfer (FTP), and e-mail
- In the late 1980’s, significant research/academic use of TCP/IP began outside the US
- In the early 1990’s, some physicists and computer programmers developed a network-browsable “hypertext” interface called the “world wide web”
- By the mid 1990’s, the world-wide web (WWW) drove a massive explosion in internet connectivity and applications; e-mail use “came along for the ride”
  - Note that the web is not the same thing as the internet – the web is one application of the internet. E-mail and instant messaging are others; there are many more that are less well-known.
- The basic WWW interface was enhanced to gather as well as distribute data
- Technology was developed to link websites to databases
- This technology allowed sales transactions to occur over the WWW
- Physical products are typically delivered by package delivery networks like FedEx and UPS, which experienced symbiotic growth with the internet during the 1990’s
  - The idea of a high-performance package delivery network using aircraft was itself pioneered by FedEx in the early 1980’s.
  - IT tools have also been critical to the growth of package delivery networks
- For “digitizable” products like software and music recordings, the product itself could also be *delivered* over the network.

Computer/network-mediated business transactions are called *e-commerce*. Note that e-commerce is very young; it barely existed in the mid 1990’s.

- Note that e-commerce does not *require* the WWW, although the WWW is a common foundation used to support e-commerce.
- For example, a “B2B” (business to business) e-commerce application in which one firm’s information system automatically communicates parts orders to a supplier’s information system would probably not use the WWW.
- On the other hand, “B2C” (business to consumer) e-commerce applications almost always use WWW interfaces (at the consumer end of the transaction).

Some common terms:

- *The internet*: the global network environment. Literally, it means a collection of interconnected networks.
- *Intranet*: the network within an organization; typically, it refers to portions of the network not accessible to those outside the organization.
- *Extranet*: one or more interconnected intranets, bridging multiple organization, but not openly accessible to those outside. For example, a firm might form an extranet with its dealers or key suppliers, in order to share critical inventory or product lead time information. This information would not be accessible or even detectable just by “googling”.

## ***Class 2 – Examples of Critical IT Opportunities and Challenges:***

Inventing or reinventing a business with the help of IT:

- Obvious examples such as Amazon.com (my own observations, not in RT book)
  - In some cases, a firm’s entire business model is based on IT and the WWW
  - Consider Amazon.com, in their original business of selling books. The key observation is that there are a huge number of different books published, and many books appeal to a “thin”, widely dispersed audience. Thus, any physical store must either limit its selection to better-selling books or carry a huge, slow-moving inventory.
  - The WWW provides an efficient way for customers to browse a huge selection (without mailing out gigantic catalogs)
  - Inventory can be concentrated in relatively few locations, where it turns over relatively quickly
  - Information systems streamline the “picking” and shipping of orders
  - Delivery via efficient package network carriers
  - Amazon.com simply could not exist without modern IT. Another example of such a business is Google.
- *Norfolk Southern Railway* (RT Example 2.2)
  - As late as 2004, Norfolk Southern (the 4<sup>th</sup> largest railroad in the US) operated its freight trains on an essentially manual basis. Dispatchers manually made up and released trains, generally with the goal of maximizing train length. While train long trains *tend* to reduce fuel and crew costs, the relationship is somewhat loose. Holding up a train’s departure can have system-wide effects that are not so easily predicted, because crews and locomotives may be needed at other locations.

- In 2004, Norfolk implemented a new information system to track its resources and make dispatch decisions in ways that consider the operation of the whole system. That would have been virtually impossible on a manual basis.
  - Financial results have improved significantly
  - Note (not in book): there is more to this story than just “IT”. It would have been possible to install a very similar-looking system that tracked all the same information but didn’t make good decisions. The success of the system depended not only on applying computer technology, but also on having sufficiently accurate mathematical/economic models of the decision/planning problems for which the system is responsible. Getting these models right is the topic of the *Operations Management* course
- *Bringing 7-Eleven out of bankruptcy* (in old edition of book)
    - Old supply system was chaotic:
      - Each store could have more than 80 deliveries per week, each with different items
      - Deliveries could occur during peak shopping hours and disrupt sales.
    - New supply system with stronger IT component:
      - Handheld computers used to place orders
      - Distribution centers consolidate each store’s orders into a single 5 AM delivery the following morning
      - Real-time sales and ordering data available to store managers and their superiors
      - Note that this application also involved redesigning the firm’s *supply chain*. It’s possible to improve supply chains without upgrading information systems, but IT can help a lot.
      - Again, there’s more than “just IT” here. Getting the inventory levels right is also an example of the kind of analysis covered in *Operations Management*. Just having a highly automated, computerized system will not guarantee that you will manage inventories well. Once you have the right analysis, then a computer system can help enormously in applying it.

System changeovers — benefits and risks: replacing outdated, outgrown, or patchwork information systems can carry tremendous benefits:

- *Sarbanes-Oxley at Blue Rhino* (in old edition of book)
  - Leading supplier of propane canisters for gas grills etc., sold and collected by independent local distributors.
  - Since market capitalization exceeds \$75 million, the recent Sarbanes-Oxley Act requires both CEO and auditors to certify the firm’s financial system and its controls
  - Accounting staff had to plug receivables and payables information from distributors into spreadsheets manually, on a monthly basis
  - Resulted in at least one week per month when inventories were not tracked accurately, so the firm had to carry an extra inventory “cushion”
    - Classic story at companies that have outgrown desktop computer tools.

- Personal productivity tools like Excel and Word are great, but they are easy to outgrow
  - If many people are repetitively using the same spreadsheet or document, or it is used for routine, cyclic tasks like logging monthly or weekly sales, you have outgrown your desktop tools, and...
  - You should invest in a larger scale IT solution constructed with database technology
- HR had to manually communicate information about new hires to the IT department (the process required manual intervention at both departments)
- Purchasing required filling out manual forms
- Sarbanes-Oxley accounting controls required improvements to all these systems; more efficient operations were a beneficial side-effect.

“Patchworks” of smaller “legacy” departmental systems can be hard to coordinate. Mergers can leave companies with a complex patchwork of systems that need to be consolidated:

- *JPMorgan Invests in IT* (RT Example 1.3)
  - Old technology and mergers had left JPMorgan with a patchwork of different systems that interfaced poorly, with 90 different data centers.
  - JPMorgan is spending \$3 billion to overhaul and consolidate its systems and networks
- *Kmart and Sears: Ignore Information Systems at Your Peril* (in old edition of book)
  - While introducing ERP systems may be nightmarish, cobbled-together groups of loosely communicating legacy systems also have serious downsides
  - Information systems should not just be an afterthought in corporate mergers and acquisitions
  - Example: Kmart and Sears merged in 2004
  - Kmart had
    - 3 inventory management systems
    - 5 logistics management systems
    - 5 supply chain management systems (the difference between “logistics” and “supply chain” is not entirely clear here)
    - 4 purchasing systems
  - Sears had
    - 5 inventory management systems
    - 4 logistics applications
    - 5 supply chain systems
    - 6 merchandise planning systems
  - Each firm’s IT infrastructure was individually a mess; now they have 37 systems to integrate.
  - IT considerations appear to be a significant obstacle to extracting value from the merger
  - While Sears/Kmart struggle with integrating these systems, WalMart and Target can press their advantage through more efficient operations and can get further ahead by adding new technology

- Moral: the right IT tools can be a key “competitive advantage”
- Consequence of moral: in many organizations, the IT department (ISD) should evolve away from just having a supporting role. Even in a business whose products/services are not directly IT-related, the IT department may need to evolve into a “partner” with (for example) marketing, finance, and/or operations.

On the other hand, IT investments sometimes do not work out very well:

- *TIAA-CREF has Problems with Upgrade* (RT Example 2.3)
  - In 2006, TIAA-CREF (which handles most university faculty and staff retirement funds), launched a major consolidation project.
  - The migration was much more expensive and difficult than anticipated; some customers had significant difficulties for almost two years
  - However, the changeover was eventually successful and many customers were not impacted
- There are plenty of ERP horror stories:
  - 1999: Hershey reported a \$19 million quarterly earnings drop when they brought ERP on line
  - 1999: Whirlpool was unable to ship large numbers of appliances after installing ERP
  - 1996/2001: FoxMeyer (a prescription drug distributor) blames bankruptcy filing on ERP introduction
  - ERP introduction often causes years of dislocation
  - Some firms simply “back out” of ERP introductions after spending millions of dollars.
- There is also no shortage of other IS efforts that have failed or have cost millions more than anticipated
  - 2002: installation of Security Audit and Analysis System (SAAS) at the IRS was a major failure. For example, once auditors entered data, they found they were not allowed access to it!
  - The FAA introduced the National Airspace System Plan (NASP) to upgrade air traffic control information systems in 1982, but had made little progress by 1991 after huge investments. In 1991, the FAA introduces a more incremental plan that has been much more effective. For example, one element of the incremental plan involved (temporarily) programming new hardware to simulate old hardware which was becoming too difficult and expensive to maintain.
- IT missteps, especially grandiose ones, can be very expensive.
- Sometimes very ambitious, far-reaching IT upgrades that sound great are too difficult to implement or introduce in practice. Gradual upgrade and consolidation of systems may sometimes work better.

Some companies do well with fairly modest IT investments:

- *Is Dollar General Really Thriving with Minimal IT?* (old version of book): Dollar General has modern IT above the store level, but has minimized its investment in IT at the store level. Individual stores have only cash registers that capture transactions and upload them to the firm's central IT infrastructure once per night. The firm has had excellent financial performance, although "shrinkage" (theft and lost goods) are bothersome and hard to address due to the lack of in-store information systems.
  - *Possible moral:* if some aspects of IT are not critical to your business case, they may not merit aggressive investment.

One clear message here is that it's important to think about IT early and try to get it right.

- Changing and upgrading systems can be really painful
- But waiting too long to put in a system is also painful; countless firms are wasting countless employee hours fiddling with spreadsheets when they should have moved to a comprehensive, multi-user database solution long ago.

There's no magic formula for how firms should approach IT, or whether a particular IT project makes sense. Near the end of the course, however, we'll discuss the "SDLC" methodology for evaluating, acquiring, and developing information systems.

I believe that basic technical understanding of the technology is very helpful to making good decisions about it. That is why we will start working hands-on with relational database software in the next class.

### ***Class 3 – Lab: Introducing MS Access tables***

### ***Class 4 – Memory storage calculations***

See *Memory Storage Calculations* handout.

### ***Class 5 – Lab: Introducing MS Access forms, queries, and reports***

### ***Class 6 – Data Management (Excerpts from RTP Chapter 4)***

We have seen a little bit now about the tables in which business data are stored, and to how to calculate the amount of storage they might consume.

Chapter 4 of RTP addresses issues of data management. Such issues include:

- How many tables should be in the database an information systems using?
- What data should be in each table?
- How the tables should be connected to one another?
- How many different information systems should an organization have?
- What should be the function of each system?
- What information should be in each system?
- How should the systems communicate?

The issues concerning interrelated and overlapping systems resemble in some ways the same questions with tables, but on a larger scale. In this course,

- We will get into a lot of technical detail about how tables should be organized within one database system. It's a well understood area and the basic concepts are not too difficult.
- As the way entire "systems" should interact, we will take a more descriptive, superficial approach.

Case illustrating some common data management issues: Example of MetLife around 1999, from old edition of book (see handout)

- The firm had a very large number of information systems, one for each product line
- Example: if a client informed the firm of a change of address when they updated their automobile policy, that change would not propagate to their life insurance policy
  - This is an example of how *redundancy* – storing the same data in more than one place – can be a headache for information systems (relate story of safety deposit box)
- Data from all these different systems was hard to integrate, making the firm hard to manage. This problem may have contributed to
  - Poor financial performance relative to competitors
  - Improper sales practices in some units, resulting in a scandal and almost \$2 billion in fines and penalties.
- The firm wanted to better integrate business units in response to legislative changes removing barriers between the banking, insurance, and securities industries.
- The company reorganized the data systems serving individual customers with products like life, home, and auto insurance. All these systems now use a common customer database.

A similar story about Panasonic is in the current edition of the textbook on pages 104-105.

MetLife's and Panasonic's problems are not unusual, and many companies have similar problems today. Generically, some common data management problems facing today's organizations are:

- The volume of data increases as new data is added. Historical data is often kept for a long time, so typically data comes in faster than it is deleted. New technologies mean that gathering new data is easier and faster. So, not only is the total volume of data increasing, but the rate at which it is increasing is also increasing!
  - Should a firm take advantage of every possible opportunity to gather data?
  - For example, any website can gather "clickstream" data: records of exactly how users moved around a website, whether they bought anything or not. This data may be of value, but can pile up quickly.
- Data tend to be scattered throughout the organization. Older organizations tend to have many "legacy" systems that communicated poorly, causing severe problems – the MetLife case is an example. Thus, it is often desirable to centralize data storage, but by no means always – it may be better to leave departments or working groups "in charge" of the data they use the most. It is costly and risky to replace older "legacy" information subsystems that are working smoothly. Replacing a large number of smaller systems with one larger one can often be very complicated and costly. Sometimes it may be better to created "federated" systems that combine information from constituent systems.

- Data accuracy – many organizations have far more errors in their databases than they are aware of. One cause is unnecessary data redundancy, but there are other causes too.
- We may also want to use data from outside the organization (either public-domain or purchased).
- It may also be advantageous to share some information with suppliers or vendors (for example, sharing information about inventories can reduce inventory fluctuations and costs throughout a “supply chain”).
- Data security and quality are important, but are more easily jeopardized the larger and more complicated an information system becomes.

We can classify the processing tasks information systems perform as follows:

- *Transactional processing* (sometimes called TPS): keeping track of day-to-day events, such as logging orders and shipments, and posting entries to accounting ledgers. In terms of a data table, transaction processing means an ongoing process of adding rows (for example, to reflect a new order), modifying table cells here and there (for example, if a customer changes their telephone number), and perhaps deleting rows.
- *Analytical processing*: means combining data from many table rows in order to obtain “higher-level” information. Entering a row into a table to reflect a newly-received order would be transaction processing; an example of analytical processing would be computing the number of orders and total dollar value of orders for this month, and comparing them to last month.
  - Analytical processing can be as simple as sorting, grouping, and summary calculations in an Access query or report. For example, providing all group managers with a summary of their groups’ costs for the month, broken down by cost category. This kind of application can be called “classic” MIS (Management Information Systems).
  - Analytical processing can get a lot more sophisticated. For example, *data mining* refers to using sophisticated statistical or related techniques to discover patterns that might not be apparent in standard reports.
  - *Decision support* can involve using the data to help managers make complex decisions, *i.e.* how to route 400 shipments from 20 warehouses to 100 customers.
  - Database systems like Access don’t ordinarily do data mining or decision support by themselves. For such uses, they usually need to be connected to other pieces of software.

Sometimes it can be mistake to do transaction processing and analytical processing on the same database, especially if the analytical processing is very time consuming or complex.

- The analytical processing may make the transaction system run slowly
- Conversely, the transactions may interfere with the analytical processing and make *it* run to slowly
- If an analytical processing step takes too long, the data it is using may change in the middle of its calculation. “Locking” the data to avoid this can block transaction processing.

It may be better to make a copy or “snapshot” of the database used for the transaction system.

- This is often called a “data warehouse” – see RTP Section 4.4.

- You can do a lot of analysis on the data warehouse without disrupting the transaction system and a lot of transactions without disrupting data analysis
- The data warehouse will not reflect the very latest transactions, but for large-scale aggregate analysis, that is probably not a big problem.
- Another reason to create a data warehouse might be to consolidate information from several different transaction systems so it can be analyzed together (see Figure 4.9, RTP page 118).

At this point, we will start getting into the details of data management for the case of a single system. You may think of this material as a very detailed expansion of the subject matter on RTP Sections 4.2 and 4.3.

### **Remainder of Class 6 – Roughly follows GB pages 218-222**

(However, I also introduced the notion of a *repeating group*.)

### **Classes 7-11 – Database design; multiple tables in Access**

Refer to class handouts.

### **Classes 12-13 – Network computing**

Note: these classes cover roughly the material in RT chapter 5 and technology guides 4-5. However, my emphasis will be different, and I'll give some details not in the text.

Broadly speaking, in terms of major impact and commercialization:

- The 1980's was the decade of personal computing (single-user, desktop computers)
- The 1990's was the decade of networking

Many of the relatively recent advances and technologies we now associate with computers are primarily networking-based.

Basic definitions:

- Something that connects two computers is a *link*
- Multiple computers connected by links comprise a *network*.
- Each computer on the network is called a *node*.
- Generally speaking, data should be able to get from any network node to any other node.
- There are many different shapes (or topologies) that can be used to build a network
  - Today the predominant network topology is a collection of interconnected "stars" or "buses"
    - "Star" – each computer has an individual connection to a central interconnection point
    - "Bus" – a group of computers share a common communication channel (either a radio frequency or a metal wire). Only one can send information into the channel at any given time.
  - At one time, interconnected "rings" were also popular.
- Some nodes on the network are specialized computers that serve primarily as connection points whose job is to make sure data gets sent to right place. Some common names:

- Switches
- Routers
- Hubs

Most of this equipment is not particularly visible to most people, although a lot of people now own wireless routers for their homes.

#### Kinds of links

- Link speed is usually measure in *bits* per second (b/s), with the usual prefixes K (kilo/1,000), M (mega/1,000,000), G (giga/1,000,000,000), etc. Note that for network speeds, it is customary to use the *decimal* form of these prefixes, not the binary ones. Unless otherwise specified, you should assume that all data transmission rates use decimal prefixes.
- Wires (usually copper) these can be used in many ways.
  - Twisted-pair wires, such as
    - Regular telephone wiring
    - Ethernet wires, which currently come in three flavors, 10 Mb/s, 100 Mb/s, and 1Gb/s.
  - Coaxial cables, like those used for cable TV. These have higher theoretical capacity but are harder to work with (they are stiff and hard to connect together).

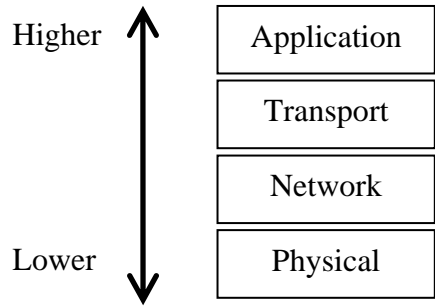
For wires, there is a trade-off between distance and the number of bits per second that may be transmitted. You can transmit with very high data rates, or over very long distances, but not both. This trade-off arises from the basic electrical resistance properties wires.
- Optical fiber (carries light pulses)
  - First commercialized in the 1970's
  - Much better than wire for combining high data rates and long distances. Links can have capacities in the many Tb/s
  - More difficult to work with than either twisted-pair wires or coaxial cables. In particular, it's relatively hard to "splice" two of them together (however, that also makes them more secure).
- Unconfined electromagnetic waves (radio/microwave /infrared/light) – "wireless"
  - Microwave links (can be directional – can be formed into a narrow beam)
  - Satellite links
  - Within-building ("wi-fi") broadcast: capacities typically 1-54 Mb/s for standard "802.11g" systems, up to 300 Mb/s for newer "802.11n" systems.
  - Mobile broadband: this technology is just emerging. Current services are advertised at \$80/month for 400-700 Kb/s downloads (but much slower uploads), with approximately the same coverage area as standard cell phones.
  - Wi-Max: see below

#### A quick history of computer communications:

- The first large-scale electronic networks built were telephone networks. But they were not used by computers initially, because computers didn't exist. In fact, "computer" was actually a job title for a person who did mathematical calculations for engineers and scientists.

- When businesses started using computers, each organization had its own computer in its own room. Data got in and out of the computer room by being physically carried as punched cards, printed reports, magnetic tape etc. (eventually floppy disks, too) – later called “sneakernet”.
- People began placing input/output (I/O) devices outside the computer room, connected by wires: printers, card readers, terminals (=printer + keyboard or screen + keyboard), etc.
- Technology was invented to encode (modulate) data into sounds the telephone network could carry. The data would be “demodulated” back into bits at the other end (thus the term “modem” – *modulator/demodulator*); see RT p. 388.
  - This allowed people to have computer terminals at home and work over telephone lines
  - Many other business applications involving sending or receiving data from remote locations
  - Early modems were slow (110 b/s = 0.11 Kb/s in the 1960’s). This rate gradually increased to about 56 Kb/s today.
  - The technology is still widely used, but in decline
- By the late 1960’s, interest was growing in large general-purpose data networks independent of the telephone network.
  - Before, such networks existed only for specialized applications (mostly military)
  - ARPANET – the (defense department) Advanced Research Projects Agency NETwork was built in the early 70’s
  - This network gradually evolved into “the internet”
  - The internet had a fairly small user base until the mid 80’s. Then it began to gather momentum
  - In the early 90’s, the “world wide web” application of internet technology became very popular and drove a massive expansion of the internet (along with the “.com boom”)
  - In the 90’s there was a general telecommunications boom of which the internet boom was a big part. Numerous firms tried to secure their place in the boom by building extensive network links, especially in North America
  - A great deal of network capacity was built. About the same time, technology appeared that greatly increased the amount of data an optical fiber could carry (by simultaneously sending multiple information streams using light beams of different colors). The result was overcapacity and a major telecommunications business “crash”. However, internet use continues to climb.

How networks work: LAYERING is very important. The four standard layers are (see also RT Figure TG4.8, p. 395):



- Bottom: physical layer – the physical workings of the links (wire, fiber, wireless, etc.)
- Network layer (typically “IP”, standing for “internet protocol”): handles the identification of particular computers on the network, and describes the structure of the network.
  - Currently, each computer has a 32 bit “IP address” (usually split into four bytes printed in decimal like 128.6.59.202).
  - The addresses have structure – for example “128.6” in the first two bytes of the address means somewhere at Rutgers (although 165.230 also means Rutgers), the 59 designates a particular “subnet” (roughly the same as a building or part of a building), and the 202 identifies which computer on the subnet.
  - Note that most computers also have a “hostname” that is easier for humans to remember, like “business.rutgers.edu” or “www.amazon.com”. While these hostnames are related to IP addresses, they aren’t exactly the same. Special computers on the network, called “name servers” provide translation between the two. Small organizations may not have a name server, relying on a name server from their internet service provider. Large organizations like Rutgers may have dozens of name servers.
  - The end of the hostname is often called the “domain name”. For example, the domain name in “business.rutgers.edu” is “rutgers.edu”. The domain name usually indicates which institution owns the computer, and what kind of institution it is. “Rutgers.edu”, for example, means the computer is at Rutgers, which is an educational institution.
  - 32 bits are no longer enough space for an IP address, and we will gradually move from IPv4 (32 bits) to IPv6 (128 bit addresses). Various workarounds suffice for now:
    - Dynamically allocating IP addresses only when computers are actively connected to the network (“DHCP” is a common way of doing this), or
    - Grouping small sets of computers to share a single IP (network address translation or “NAT”). For example, if you have a router in your house or dorm room, it is probably doing NAT translation, so that all the computers in your house/room effectively share a single IP address.
- Transport layer (typically “TCP”). Specifies the structure of the information being moved across the network
  - TCP specifies up to 64K (binary K) logical “ports” for each computer on the network. Each port is typically used for a different application. For example, standard web page viewing usually uses port 80.

- For each port, there may be one or more “sessions” or logical connections between to computers. For example, you could have two independent web browser windows connected to the same website from your own PC; each of these windows would represent a different session for TCP port 80 on the server. Each session is initiated by one computer sending a request to another. If the second computer agrees, the session is opened, and data may then flow in either direction.
- For each session, there may be a sequence of messages in each direction
- TCP is a “packet switched” protocol – messages are cut up into “packets” that might take different paths through the network and are reassembled at the destination (see RT Figure TG4.9, p. 396; by contrast, traditional telephone networks are “circuit switched” – the whole conversation uses the same route through the network). The size of packets varies by the application and network hardware in use, but they are typically roughly on the order of magnitude of 1KB.
- Application layer: specifies different protocols to move data for various uses. These protocols constitute an “alphabet soup”:
  - First: TELNET (old) – run a terminal session (a back-and-forth text-based interaction between a person and a computer – kind of like the “command prompt” in Windows)
  - FTP (old) – move files back and forth (still in some use when security isn’t an issue)
  - SSH – (“secure shell”) encrypted terminal sessions and file transfers. This accomplishes the same basic tasks as TELNET and FTP, but is far more secure. Other protocols can be “tunneled” through SSH to make them secure.
  - HTTP/HTTPS – hypertext transmission. This application appeared in the early 1990’s and rapidly evolved into a way of projecting a wide range of graphic user interfaces across the internet. The “S” in HTTPS means secure/encrypted; otherwise HTTPS is identical to HTTP. HTTP/HTTPS is a much easier and more secure way to do arbitrary things on a remote user’s screen than making the user run custom software.
  - SMB, NFS – file sharing. Makes disks on a distant computer look like they’re on yours (with the correct software, you can also exploit SSH for such purposes).
  - SMTP – sending e-mail to and between mail servers (computers that can route e-mail). This is a “push” protocol: the computer requesting the connection sends the messages; when your computer sends an e-mail, it typically uses SMTP.
  - POP3, IMAP – retrieving mail from e-mail servers. These are “pull” protocols: the computer requesting the connection receives the e-mail messages (if there are any)
  - And many, many, more...
- Typically, each protocol uses a single TCP port (or perhaps a few). For example, HTTP usually uses port 80, and SSH usually uses port 22.

Some more notes on layers and protocols:

- The standard URL (“Uniform Resource Locator”) used by web browsers has the basic form

*PROTOCOL://hostname/additional-information*

Thus, <http://www.rutgers.edu/> means “HTTP protocol, computer with hostname [www.rutgers.edu](http://www.rutgers.edu/), no additional information”. A numeric IP address can also be used

instead of a hostname, as in `http://123.57.12.92/obscure-stuff` (but is a clue that the site may not be safe to connect to).

- As you move downwards in the protocol layer “stack”, more and more “bookkeeping” data — also called “headers” — get appended around the data you are sending. This means the actual number of bits transmitted can be substantially more than you might think. For example, when a message is divided into packets, information is added to each packet so that it gets routed correctly and the packets may be reassembled in the right order.
- TCP and IP are usually used together and are known as “TCP/IP”
- You can run more than one network layer on top of a physical layer on the same link (for example, IP and AppleTalk)
- You can run several transport layers on top of a network layer (for example, TCP and UDP on top of IP)
- And, of course, you can run many application layers on top of a transport layer (SSH and HTTP on top of TCP)

#### Kinds of networks

- LAN – “Local Area Network” on the level of a single building, part of a building, or office
- WAN – “Wide Area Network” a somewhat vague term for a network that covers a “non-local” geographic area, that is, something larger than a LAN. An example would be the network that connects Rutgers’ various campuses (including Camden and Newark).
- Enterprise network – refers to the totality of an organization’s networks, both its LANs and its WAN(s) together.
- Internet – multiple networks connected together
  - The idea of *an* internet preceded the current notion of *the* internet – “the” internet came into existence when almost everything got connected into one huge network!
  - The “IP” network layer was specifically designed to make it easy to create internets. That is why all internets essentially merged into “the” internet that grew so quickly in the 1980’s and 1990’s, and conversely why IP is now the dominant network layer.
- Intranet – the portion of an organization’s enterprise network that is not accessible by arbitrary internet users
- Extranet – when multiple organizations connect their networks in a way that is not fully accessible from outside that set of organizations
- VPN – “Virtual Private Network” – an intranet or extranet that physically uses the general internet, but is encrypted in such a way that it looks like a private WAN that outsiders can’t snoop on (we hope).

#### Current network technology

- Most firms now have LANs implemented with copper wire, usually Ethernet, and now also building-level wireless
- Many larger firms have WANs containing wire and/or fiber and maybe some satellite or microwave links (depending on the firm’s size). The longer links in these networks are typically leased from ISP’s (see the next item)

- *Internet service providers* (ISP's) are firms maintaining interconnected, overlapping networks made primarily of fiber (examples: AOL, ATT, Sprint, etc.), but possibly also involving wire, satellite, and microwave links. ISP's also lease capacity to firms for use in WANs. Large- and medium-size firms connect directly to ISP's.
  - Also, there are some non-profit alternatives to ISP's, like "Internet2", a consortium of large universities like Rutgers, and other large nonprofit organizations.
- Large firms can afford to lease dedicated high-speed connections to ISP's, like "T3" lines
- The dreaded "last mile": smaller firms and individual households connect to the ISP's in various (often non-ideal) ways:
  - By phone and modem (sometimes directly to an employer instead of ISP); this method is becoming increasingly rare
  - Cable modem – signals carried over the same coaxial cable that distributes TV signals. Capacity usually 0.5-7.0 MB/s, but capacity may be shared with other users in the neighborhood
  - DSL – signals carried over regular phone lines, but not at audible frequencies. About 0.5-1.0 Mb/s, but occasionally faster. Only works if you are within 2 miles of telephone switching center, but does not have the same capacity sharing issues as cable modem technology.
  - Direct fiber connection (example: Verizon FiOS) – run optical fiber all the way to individual houses or small business. This is the most straightforward technology, but is being rolled out slowly in limited areas. Depending on the price paid, individual households can get 5-20 Mb/s download speeds and 1-5 Mb/s upload.
  - WiMax – (see RTP p. 212) a wireless networking technology with a theoretical reach of up to 70 miles and transfer rates up to 70 Mb/s. However, there is a tradeoff between distance and speed, and more typical performance would be about 10 Mb/s over 6-7 miles, if a "line of sight" connection is possible. This technology is rare in the US and is currently targeted at areas lacking wired infrastructure (especially rural areas). WiMax is primarily conceived as a wireless solution to the "last mile" problem, and not a mobile technology for battery-operated devices such as laptops.
- Most network connections carry a fixed charge per month, without tracking the exact number of bits sent – one reason we have so much "spam"!

Uses for networks are expanding all the time. For a fairly current catalog, see Sections 5.1-5.2 of RT.

Data transfer calculations – how much time will it take to move a data file? (The file could contain text, data tables, video, audio, etc.)

- Calculate the amount of data to be moved in *bits*
- Divide by the speed of the transmission line in bits per second
- Convert from seconds to larger time units like minutes or hours if necessary
- Remember:
  - File sizes are commonly in bytes, most often with *binary*-style K, M, G etc.
  - Transmission line speeds are usually in *bits* per second, with *decimal*-style K, M, G etc.

- This mismatch is annoying, but is the common convention.
- It's easiest to convert the file size to decimal bits, and then divide by the line speed.
- The protocol stack will add header information that will cause the real download to take longer in practice
- Network congestion or transient malfunctions could cause even more delays

**Sample file transfer calculation:** Suppose we want to do “video-on-demand” downloads of 4 GB movies in DVD format (binary-style GB). How long would that take over a 1 Mb/s DSL line?

$$\text{Size of movie} = (4 \text{ GB})(1024^3 \text{ B/GB})(8 \text{ bits/B}) = 3.44 \times 10^{10} \text{ bits}$$

$$\begin{aligned} \text{Seconds to transfer with DSL} &= (3.44 \times 10^{10} \text{ bits}) / (1 \times 10^6 \text{ bits/sec}) = 3.44 \times 10^4 \text{ sec} \\ &= (3.44 \times 10^4 \text{ sec}) / (60 \text{ sec/min} \times 60 \text{ min/hr}) = 9.54 \text{ hours} - \text{probably not acceptable!} \end{aligned}$$

Note that actual transfer times would be somewhat larger due to overhead (headers) added by the application, transport, network, and physical network layers, and because of possible network congestion.

## **Classes 21-23: Ethics (from RT Chapter 3)**

### **Ethics...**

- ...is the branch of philosophy concerned with right and wrong (which I won't try to define here!)
- Ethics and legality should not be confused with one another, although they are (we hope!) related

The existence of modern information technology raises some ethical issues (RT Section 3.1, pp. 62-68; in particular, see the checklist on RT p. 63):

- *Privacy*
  - It is becoming increasingly feasible and cost-effective to gather huge amounts of data, especially on individuals
  - What kind of data should be gathered? How long should it be kept?
  - Are individuals aware of what kind of data is being gathered?
  - Who controls this information?
  - How is it used or shared?
  - How secure is the gathered data?
  - How much should employees be monitored?
  - What kind of information should be made widely available?
- *Accuracy*
  - How accurate is your data?
    - Consumers Union consistently finds that credit reports maintained by information brokers like TRW contain errors.
  - Who is responsible for making sure information is accurate? Can injured parties seek compensation for damage caused by inaccurate data?
  - What are the sources of inaccuracy? Are they intentional or accidental?
  - Is new technology making it easier to distribute false or distorted information?
- *Property*
  - Who "owns" the information? What does it mean to "own" information?
  - What are "fair" prices for information exchange?
  - How does one deal with information piracy? Is piracy justified if information prices or ownership are "unfair"?
  - Should employees use corporate information infrastructure for personal purposes?
- *Accessibility*
  - Who has access to information?
  - Is technological change benefiting or hurting particular groups (for example, people with disabilities, people with lower incomes, people in specific geographical regions) by changing their (relative) degree of access?

Expansion of discussion of privacy and property:

### *Privacy:*

- Evolving technology is making it easier to amass ever-growing amounts of data about individuals

- This information is often sold and exchanged by organizations without our knowledge, sometimes intentionally and sometimes unintentionally
  - Massive TJX (T.J. Maxx, Marshall's, Bob's...) data breach revealed in 2007; see RT pp. 60-61. Not only was data compromised, but the company was keeping credit card data it was supposed to delete once transactions were completed.
  - JetBlue violated its own security policy in 2003 by transferring data on 1.5 million customers to a security contractor. This contractor matched the data with other sources to add social security numbers, income, occupation and other data (in old book)
- Is a reasonable degree of personal privacy being eroded?
- Corporations are now required to have *privacy policies* that are distributed to their customers
  - These are dense, long, legalistic documents. Does anybody read them?
  - How about regulatory standards? Fairly weak at present. A bureaucracy/agency to monitor standards compliance would of course have its own drawbacks.
- Combining data from various firms and/or government agencies can allow extremely detailed data to be synthesized
  - At the most basic level, acquire data tables from different sources and then do "join" operations on them
  - *Data aggregator* firms like ChoicePoint (see RT p. 64) specialize in this synthesis
  - In 2004, ChoicePoint fell victim to a simple scam, selling very detailed data on 150,000 people to identity thieves (in old book)
    - The thieves set up over 100 fake companies that each bought relatively small amounts of data from ChoicePoint. Then the thieves did their own join and union operations to assemble a large volume of identity theft data.
    - See RT p. 67 for a security breach incident involving another aggregator, LexisNexis (but this was more of a classic security incident)
  - Data exchange can also have benefits: inadequate data exchange between US government agencies is often blamed for allowing the 9/11/2001 attacks.
  - What is the right balance?
- Monitoring: evolving technology means that we may be monitored more than we might expect from past experience.
  - Security cameras (in the UK, for example, there are estimated to be some 4 million security cameras in operation)
  - If you carry a mobile phone, your movements can be tracked quite accurately as long as it is on; you do *not* have to be making a call
  - Keystroke monitors – can capture every action an employee takes on their computer/terminal (can also be planted by hackers)
  - Do truck drivers, police, and taxi drivers want GPS units reporting where they are every second? (Recent conflict in Philadelphia.)
    - GPS-based car insurance rates?
  - It is legal (but not always expected) for companies to monitor/read their employee's e-mail. The rationale is that the companies own the infrastructure through which employees send and receive e-mail. For some people, it may be important to maintain separate work and personal e-mail accounts. Is there a

good reason e-mail communication should be less private than telephone and postal communication?

- It is common for organizations to keep extensive records of internal e-mails, because they are considered “business documents” (unlike much of the communication they replace, like phone calls and face-to-face conversations). E-mails have become an important source of evidence in prosecutions, lawsuits, and the like.
- Unexpected information release
  - Certain kinds of information (like campaign contributions, property deeds, salaries of public employees) are part of the public record
  - Ordinarily, they are available, but require some effort to get at
  - Some organizations collect this information and either sell it or even make it freely available on the web
    - Example: fundrace.org (in old book) – you can view the campaign contributions of all your neighbors (and see their occupation too)
- New information technologies evolve through innovation in the commercial sphere (although often based on basic research from universities etc.).
  - Lack of precedent and ownership of infrastructure means tend to mean that corporations tend to have the legal upper hand in privacy conflicts with consumers.
  - In the past decade, there was a political aversion to imposing new business regulations; collective security often seemed to take precedence over privacy in governmental management of information, too.
  - How much of this situation is necessary to promote innovation/security?

*Property:* issues regarding ownership and distribution of information

- Trade secrets: information that firms intend to keep confidential or share only with business partners who have agreed not to disclose it.
  - Modern information technology makes trade secrets easier to steal; however, this is primarily a *security* issue (discussed later)
- Copyright and distribution issues: concerns information-based products that firms sell to their customers
  - Examples:
    - Text and graphical material (books etc.)
    - Films and videos
    - Music
    - Database resources
    - Software
  - In the past:
    - Such information was more strongly tied to physical media
    - Physical media were relatively expensive, slow, and/or difficult to copy
      - Quality of copies might be poor
      - It might be hard to make large numbers of copies
      - Copying equipment required major capital investment
      - In some cases, copies could be traced

- Copyright laws have been instituted, dating back as far as 1662, to protect books from massive copying
  - Copyright law augmented at various points in the 18<sup>th</sup>, 19<sup>th</sup>, and 20<sup>th</sup> centuries
  - Since traditional printing presses are large and fairly easy to trace, such laws were mostly adequate for about 300 years.
- Modern information technology has altered the situation:
  - Information more easily separated from the physical medium
  - Can be stored on hard disks etc. and transmitted over high-bandwidth networks
  - Modern input and output devices make quality, hard-to-trace physical reproduction feasible at low cost
    - CD and DVD burners
    - Laser printers
    - Scanners
- Result: massive “piracy” of copyrighted material in some areas
  - Music
  - Film/video
  - Software
- Copy protection technology is only partially effective. In principle, information that reaches the user in unencrypted form can always be copied.
- Piracy uses both physical media and networks (sharing sites like Kazaa, Napster, etc.)
- Music and text/graphics may now be distributed very effectively without the blessing of a mainstream “publisher”. Video will reach that point soon. This phenomenon raises the issue of whether publishers remain necessary.
  - They act as gatekeepers or certifiers of content quality. But how reliable?
  - They can provide marketing/promotion resources
  - They still control the primary physical distribution channels, but there are now effective competing channels
- But how to ensure that musicians, authors, filmmakers etc. are paid for their work? Creating “content” still requires talent and a lot of labor (especially film/video)
- High content prices may not be sustainable in the new environment

### ***Classes 23-26: Security (also from RTP Chapter 3)***

Modern information technology has made it much faster, easier, and cheaper to

- Store
- Move
- Organize
- Manipulate/process

... information than with older “manual” technology.

Unfortunately, the same technology can also make it faster, easier and cheaper to

- Abuse

- Corrupt
- Destroy
- Distort
- Falsify
- Steal

... that same information!

The internet magnifies the problem because millions of unsecured or partially secured computers are now in potential communication.

An electronic business environment also makes impersonation of others – identity theft – relatively easy.

Basic security terminology:

- *Threat*: some danger to proper operation of your information systems.
- *Exposure*: something bad that could happen to your information systems. General categories of exposure include (for example)
  - Loss of data
  - Improper release of data
  - Disruption of normal operations at your organization
- *Risk*: likelihood that something bad will actually happen
- *Controls*: procedures, devices, software etc. that you put in the system to reduce risk
- Hypothetical illustrative example:
  - A hurricane is a potential threat
  - Your data center's exposures to hurricanes consist of disruption of normal operations, and possible loss of data (due to flooding and power failures)
  - Your risk depends on how likely hurricanes are at your location
  - The corresponding controls might consist of anti-flood pumps, plus a backup power system connected to both your computers and the pumps. Another control might be regular off-site backups of your data.

“Risk analysis” process:

- Assess information assets and their exposures
- Estimate probability of each threat
- Consider how to *mitigate* risk via controls
- *Evaluate* cost effectiveness of controls

Risk mitigation strategies:

- *Acceptance*: “live with it”. Examples:
  - A data center in the Nevada desert might choose not to protect itself against floods – they are too unlikely to justify a large expense to protect against
  - You might accept the risk that your top corporate officers could misuse sensitive information – they also have a compelling legitimate need for the information, so that need could outweigh the risk.
- *Limitation*: implement a control that reduces the risk (try to make cost of control proportional to risk)
- *Transference*: transfer the risk to somebody else – for example, buy insurance

There is no such thing as “total” security:

- Don’t think of security issues as “one-time” problems; it is an ongoing process and a portion of the workforce needs to be dedicated to it
- Need to consider security-related costs when looking at cost-effectiveness of computer technology
- Some otherwise good ideas for IT projects might be too risky/difficult to implement securely
- With awareness and effective countermeasures, security can *usually* be manageable

Unintentional threats (accidents):

- Accidents always were a threat to organizations’ data. Fires and hurricanes can destroy paper files just as easily as computer files
- Centralized systems can be vulnerable to problems at the central site
- Distributed systems can be vulnerable to problems at just one site (if their design lacks suitable backups/redundancy)
- Power failures can do a lot more damage than they used to
- With the introduction of computers, there are a lot of new ways for things to go wrong
  - Hard disk “crashes”
  - Software “crashes”
  - Software “bugs”
  - Etc...
- “Human” error and unintentional failure to follow procedures has always been a problem, but can now have more “leverage”
- Loss of misplacement of mobile devices such as laptops, PDA’s, blackberries, and thumb drives (also: unintentional damage to such devices)
- Countermeasures/controls:
  - Backup, backup, backup, backup, backup
    - Data backup
      - Redundancy is undesirable *within* a transaction database, but it is good to have a recent backup copy (or effective equivalent) for the *whole* database.
      - Can restore the database from a recent backup and a log of recent transactions
      - Can back up data to external media (CD-R, DVD-R, tapes) – protect the media!
      - Unfortunately, hard disks have been growing much faster most other media – external hard disks a good choice for PC’s now (store in fireproof box or offsite).
      - Back up data to another site over a network
    - Power backup devices (generators, “uninterruptible” (battery-supplemented) power supplies [UPS], etc.)
    - Backup of software
    - Have a backup plan for entire hardware system (rent replacement hardware, for example)

- For software developed in-house: proper development, maintenance, and lifecycle procedures to contain damage from bugs (discuss later in course)

Remaining threats are intentional – caused deliberately by people

- Internal to your organization
- External to your organization
  - People/organizations you would ordinarily have contact with
    - Partners
    - Vendors
    - Customers
    - Contractors
  - People you wouldn't otherwise have contact with, generally thieves and vandals

Internal threats and problems – employees and consultants

- The larger the organization, the higher the frequency of
  - Employee mistakes or failure to follow procedures
  - Dishonest employees (rarer, but still a concern)
- Shortcuts or dishonesty by MIS employees may have a lot of “leverage” and may be hard to detect
  - “Trap doors” – some unofficial way to gain access to your system
    - Could have been created by an IT employee to make it easier to work offsite
    - Or for some malevolent reason
  - “Skimming” – classic story of rounding down all interest payments to bank accounts, with the extra pennies going into a programmer's account!
  - “Logic bombs” – secret mechanisms that can sabotage a system. Could be either
    - Active – example: if your server receives a particular command sequence over the web, it deletes critical data and shuts down, or
    - Passive – for example, if the employee does not log in for at least two months, the system deletes critical files.
  - ...
- Countermeasures/controls:
  - Separate functions: for example, most programmers shouldn't have access to real customer data
  - Use data access hierarchies and rules
  - Controls on what data may be taken offsite and how
    - Restrict use of floppy drives, CD/DVD burners, even USB ports to prevent sensitive data being moved offsite (for USB ports, this is called “podslurping”)
    - For individuals allowed to take sensitive data offsite, make sure it is properly encrypted
  - Store data in encrypted form so only authorized users may read it
  - Monitoring (this can take many forms, and has ethical drawbacks)
  - Support your employees – make it easy (or automatic) for them to do backup, install security software etc.

Some internal threats involve employees taking/selling sensitive data outside the firm. These threats are very similar to external threats:

External threats – business partner, vendor, and customer issues:

- If you interact electronically with vendors, customers, and partners, you may be exposed to their security problems as well as your own
  - Example: LexisNexis breach, RT p. 67
- Exacerbated by recent “outsourcing” and cooperation trends like
  - EDI (Electronic Data Interchange): firms automatically share data they believe are relevant. For example, we may let our suppliers see our parts inventories so they can plan better
  - ASP (Application Service Providers) – outsourcing of specific applications such as payroll
- Web commerce technology can make improper/questionable monitoring of customers practical/profitable. Examples include cookies (see RT p. 78) and web bugs
  - Cookies are mini-files maintained by your browser under control of remote sites. They allow websites to “remember” information about you, so you don’t have to re-enter it every time you visit the site. But they can be manipulated to monitor your behavior (to varying degrees, depending on your browser’s security features)
  - “Web bugs” are invisibly small “graphics” in HTML-formatted e-mails. If you view the e-mail, the planter of the bug can tell you received it, and your IP address. That is why many mail programs now ask you first before showing the graphics in HTML-format mail.
- In an e-business environment, it may be harder to tell legitimate vendors, customers, and partners from crooks masquerading as such. Vendors and customers are harder to impersonate in person or even over the phone.
- Countermeasures?
  - Limit access
  - Investigate partners
  - Try to use reputable vendors/partners
  - Encryption
  - Consumer awareness

Other external threats

- Two motivations
  - Personal gain – thieves
  - Malice/troublemaking – hackers etc. (harder to understand)
- These threats always existed, but computer technology – and especially network technology – makes attack much cheaper, faster, and easier
- Various forms of undesired software
  - Pestware – software that tries to take over functions from your preferred software, or makes itself hard to dislodge (the architecture of Windows was designed to hide key functions from obvious view, and is therefore very pestware-friendly)
    - Adware – software that “pops up” undesired advertising in various forms; this is the most common form of pestware and usually relatively benign
  - Malware – malevolent software that sneaks into your computer

- Spyware – software that sends information from your system to others without your consent
    - Snoopers and sniffers: monitoring networks as others’ data passes by (especially passwords)
      - Wireless networks especially vulnerable, if not encrypted
    - Keyloggers (used in LexisNexis incident) – record all your keystrokes
    - “Screen scrapers” – capture contents of screen
    - Programs that look for files that may contain personal information
  - Spamware – subverts your system to send spam
  - Pestware/malware may get placed on your computer through malicious websites or e-mail attachments
- Hacking: gaining access to private systems and data (and possible abusing/damaging them)
  - Port scans: try to connect to a large number of different TCP port on a single target system. Which ports respond and how give a “fingerprint” of the kind of system and what software it has
  - Bug exploitation (usually in operating systems, browsers, and e-mail programs).  
Example:
    - A website has a form which allows people to enter some data (a review of a movie, for example)
    - You find out/suspect the web server software has a “buffer overrun” bug
    - You send a “movie review” that is much longer than the “buffer” reserved for it on the web server
    - Your “review” actually contains a machine-language computer program
    - Your “movie review” overwrites part of the web server program
    - If that part of the program happens to get executed, your uploaded program is now in control
- Spam
  - Time-wasting
  - Nowadays, contents usually criminal/dishonest/fraudulent
  - “Social engineering” and “phishing” – faking messages from tempting or official sources to induce people to run booby-trapped software, reveal passwords, or disclose other confidential information
    - Recent development: “spear phishing” – selecting a specific person so that you can send a more convincing phishing attack.
  - Unintended consequences: would the inventors of e-mail have guessed that the vast majority of all e-mail would eventually consist of unsolicited offers for fraudulent loans, prescription drugs without prescriptions, get-rich-quick schemes, stock pump-and-dump campaigns, and impossible anatomical “enhancement”? Unfortunately, the cost of sending spam is too low.
  - Legislation (“Can Spam”) has not been effective in reducing spam
    - Not a priority for law enforcement agencies
    - Spam sources often in other countries from recipients

- New communication media spawn new kinds of spam – for example instant messaging (IM) spam, called “spim”.
- Annoyance/vandal attacks – denial of service (DoS)
  - For example, bombard a server computer with bogus messages so it has no time or network capacity to do its real job
  - Often executed through “botnets” – groups of computers that have been infected by “malware” allowing malicious control
- Self-replicating attacks: viruses and worms
  - Once present on one system, they try to use that system to propagate themselves to other systems
  - May move via e-mail and have a social engineering aspect (like much spam)
  - But may exploit a software security hole (like a forgotten trap door) and not require any human participation (this mode of propagation is rarer)
  - Can reproduce very quickly if human participation isn’t needed
- The more powerful software is, the more vulnerable (MS Office macros)
- Trojan horses: software that appears to have one function but has a hidden agenda. For example, a free DVD player program that sends personal data back to an identity thief
- Phishing – gathering personal information under false pretenses
  - Example: e-mails supposedly from your system administrator asking for you to “confirm” your personal information or password
- Pharming – set up a website that looks like a legitimate business, and use it to gather personal data. The business may be made-up, or the pharm may masquerade as (“spoof”) the site of a real business
  - In 2007, somebody discovered a problem in TCP/IP that would have allowed massive redirection of traffic from legitimate websites to fake ones; *i.e.* pharming on an unprecedented scale. The software of all DNS servers in the world had to be “patched” in a very short time frame
    - Once a security patch is made public, hackers can analyze its contents and figure out what vulnerability it fixes
    - Thus, once the patch becomes public knowledge, systems without it become vulnerable
    - So, must propagate patch quickly!
- Many attacks combine categories. Examples:
  - spam + phishing + pharming: spam tells you that you need to update your personal information stored by a large company like Amazon, E-Bay, or CitiBank, and then directs you to a fake version of their website
  - spam + virus + spamware: spam has an attachment that takes over your computer and sends the same spam to everybody in your address book
- Hierarchy among hackers and spammers
  - “Script kiddies”
  - Spam pyramid schemes?
  - I receive many copies of essentially the same spam, purportedly from different people. Often the pictures in hundreds of different spams are identical.

## Countermeasures:

### Controls

- User identification and *access controls*
  - Passwords
    - Make sure they are not vulnerable to guessing (see RT p. 83)
      - To prevent computer password attacks that involve a trying a long list of possible passwords, most systems now force a delay period between rejecting an incorrect password and accepting another password.
      - Many systems require passwords that have a mix of letters and numbers, a mix of upper and lower case, and are not dictionary words. Some now also require special characters like #, %, @, etc.
    - Have a change schedule
    - Problems:
      - You accumulate too many passwords
      - So, you have to write them down or use one password for several systems – so compromising one system can compromise others
      - Vulnerable to snooping interception with some older protocols like TELNET and FTP
  - Password generators: small electronic card that combines
    - Fixed user password
    - Internal passcode
    - Time
  - ... to produce a password with a very limited lifetime
    - Example: “SecurID” and “CryptoCard”
  - Biometrics: promising, but:
    - Expense?
    - Reliability?
    - Technology sufficiently mature?
    - Fingerprint readers were common on laptops for a while, but I suspect they were not used much
- Access controls within a computer system (server)
  - Read, write, execute, (delete) privileges for files or blocks of information
  - Basic levels: user/group/all
  - More advanced: hierarchies and “access control lists” (ACL’s). Hierarchical granting of access – used for example for university class registration data:
    - Registrar can see all registrations
    - Registrar grants access to deans to see registrations for their own schools
    - Deans grant access to department chairs to see registrations for their own departments
    - Department chairs grant access to instructors to see registrations for their own courses
- Restrict physical access (physical controls)
  - Example – US Government systems with classified data are supposed to have no physical connection (other than power cords) to any unclassified system.

- If a computer seems compromised by hackers or viruses, physically detach it from the network immediately
- Some physical controls have limitations (for example, “tailgating” – somebody places their ID in a scanner, an automatic door opens, but somebody follows in close behind)
- Audits/verification
  - Example – user-verified paper voting records (how else could you do a “recount” of an election involving electronic voting machines)?
  - IT audits – performed by IT units within major accounting or consulting firms (examples: Deloitte, Accenture)
- Scanning software and hardware
  - Virus scanners: scan received disk files and arriving e-mail for suspicious patterns
  - Spam filters
    - Many use a Bayesian statistical method: use
 
$$P\{\text{message contains “Viagra”} \mid \text{it is spam}\}$$

to help determine

$$P\{\text{message is spam} \mid \text{it contains “Viagra”}\}.$$

Here, the symbol “|” denotes “given that”.

      - Spammers try to confuse these filters by misspelling key words and including large amounts of random text.
  - Watch network for suspicious packet patterns
  - Other forms of monitoring (again, how much is acceptable?)
- Firewalls (hardware and software): block traffic into your network or computer
  - Example – for a home, do not allow any connections initiated from outside; this mode is the default for most home-style routers
  - Example – for medium-sized business, block all incoming connections except SMTP (incoming e-mail) and HTTP (people trying to look at your website) into your mail and web servers (respectively).
- Virtual private networks – use encryption to simulate a private network even if parts are carried over the internet (or some less secure private net)
- Encryption!
  - Encode network traffic so bystanders cannot snoop
  - Can also be used for files on disks, USB memory keys, etc.
  - Unintended consequences: also very useful for criminals
  - Will not discuss technical details of encryption here – discussions in most MIS textbooks are oversimplified.

## **Classes 27-28 – Information Systems Acquisition**

There is a spectrum of ways for firms to construct their information systems:

- At one end of the spectrum, firms can purchase (or lease) “off-the-shelf” or “turnkey” systems, with some minimal configuration
- At the other end of the spectrum, firms can develop their own systems “in-house” or “from scratch”, doing a lot of computer programming

There are many possibilities between these two extremes; examples:

- Buying an off-the-shelf system and adding some custom functionality
- Buying major software modules from various suppliers, and connecting them together with some custom programming.

In practice, nothing is *totally* “from scratch”. Almost any corporate IT project will use

- Mostly standard hardware
- Standard operating systems like Windows or Linux
- Standard database management tools like Access, MySQL, SyBase, or Oracle
- Standard programming languages like C++, Java, JavaScript, Visual Basic
- Etc...

In addition, customized work can also be outsourced: a firm can hire another firm to

- Write customized software, or
- Customize/configure existing software
- Connect standard software modules in a customized way
- Etc...

We use the term “acquisition” to mean any process along this spectrum, from simply buying, to a mostly “from scratch” programming project, and any combination of in-house and outsourced work. “From scratch” work is also called “development”.

RT (Section 10.1) suggests that:

- New information systems should be justified by cost-benefit analyses. In practice, that may be hard to do rigorously.
- New information systems should be “aligned” with the organization’s “strategic plan” – but strategic plans can be very vague, and “alignment” hard to define.

The key points here are that resources for IT acquisition may be limited, so firms should try to prioritize IT projects by:

- Their payoff to the organization (including how closely they are related to any defined strategic goals)
- The amount of effort required.

It’s hard to focus on a lot of complicated projects simultaneously, so it is helpful to have a process that keeps too many projects from going forward at once (see for example the British Telecom example, RT p.299). Adoption of any new system, even if it’s “off-the-shelf”, should be considered a “project”.

Acquisition of new systems often costs much more than expected and can sometimes fail spectacularly – examples from the US government (including the FBI, FAA, and IRS) are well known, but there are many private examples too.

The most proven management process for avoid repetition of “classic” acquisition mistakes, is called SDLC – *System Development Life Cycle*.

SDLC is a cascade or “waterfall” of stages; see Figure 10.2 on RT p. 302.

- I have seen descriptions ranging from 5 to 8 stages. There are many variations in the exact number of steps and their names. Most critically, there are two stages at or near the beginning called “analysis” and “design”.
- Each step has a “deliverable” on which all interested parties “sign off”. In the early stages, this deliverable is a document, most importantly a “specification” and a “design document”. Later, the deliverable may be some version of the system itself.
- If problems are found at any stage, you go back to the previous stage, or perhaps back more than one stage. But the idea is to plan ahead at each stage to reduce the probability of having to go back later, and the severity of the issues that might have to be revisited.

A 6-stage version:

1. Feasibility and planning (called *investigation* in the book)
2. System analysis
3. System design
4. Programming (also called “implementation”)
5. Cutover (sometimes also called “implementation”, just to confuse things)
6. Maintenance

The following description assumes that the project involves a significant amount of custom programming or system configuration. For predominantly “off-the-shelf” adoption projects, some of the steps below can be greatly condensed.

Step 1: *Feasibility and Planning* – identify the problem and the general form of the solution

- Identify problem to be solved
- Determine goals
- Evaluate alternatives
- Examine feasibility
  - Technical: do the necessary technology and skills exist? Do we have access to them?
  - Economic: will it be cost effective to develop/acquire the system? Will the system be cost effective in practice?
  - Organizational: will the system be compatible with the organization’s legal and political constraints (both internal and external)
  - Behavioral: will the people in the organization accept the system? Will they be likely to sabotage, override, or ignore it? What kind of training and orientation will be necessary? How will they be likely to use it? Are we attempting technical fix to an organizational problem that would be best addressed another way?
- Sometimes this step can be merged with the systems analysis (next) step

Step 2: *Systems Analysis* – specify exactly what the system will do

- Define inputs, outputs, and general methodology
- Create basic conceptual structure

- Specify in detail how the system will look to users and how it should behave
- Possibly construct dummy screens/forms and reports, or even prototype systems
- Leads to a *requirement* or *specification* document. This document should be “signed off” by the parties involved, especially those who will use the system.

Step 3: *Systems Design* – say how you will meet the specification

- Describe as collection of modules or subsystems
- Each module may be given to different a programmer or team
- Design specifies how modules will communicate (inputs, outputs, etc.)
- Can use specialized/automated design tools
- Can build prototypes
- Leads to a *design document* – a description of how you will create the system. Managers and programmers “sign off” on this document.
  - Many “computer people” like writing code but not documents, so they may resist this phase
  - *But* it is much cheaper and easier to catch big mistakes in a design document than after you’ve started writing a huge program or bought an off-the-shelf product that can’t easily do what you want.

Step 4: *Programming* – build the system! (Also commonly called *implementation*.)

- Test subcomponents thoroughly as you create them
- Make *unit tests* to exhaustively test each module before connecting modules together
- Some firms have a separate group of “QA developers” to test things again, possibly with help from future users. In the book, this sub-step is depicted as a separate stage called “testing”.

Step 5: *Changeover* or *cutover* – start using the new system (sometimes also called *implementation*, just to keep things confusing)

- *Crucial*: final testing before cutover
- Cutover can be really painful, especially if the old system was already automated
- Options:
  - “Cold turkey” – do it all at once; very risky
  - Parallel – use both systems at once
  - Phased – gradual
    - By part of system
    - By part of organization (regions, departments)
    - Can be difficult to implement; allowing for phased cutover may complicate the design of the new system and may require improvements to the old system (even though you plan to stop using it soon).
- Not unusual for organization to “roll back” to an old system (and maybe try again)
- Cutover is much easier if users were already “on board” in specifying the new system
- Preparation/training might be crucial in some cases

Step 6: *Maintenance* – fixing problems, adding features

- Except in emergencies, it's best to collect sets of changes into a *release* which can be thoroughly tested
- Install new releases periodically; not too often
- Develop a “QA suite” or set of *regression tests* to check that bug fixes don't create more problems or revive old bugs (“rebugging”)
  - Expand QA tests as features are added

It is critical to involve eventual system users in decision-making in most stages (typical exceptions: programming and design).

Outsourcing of programming work may obviate a firm from having to follow some of the SDLC steps, but it does not exempt a firm from *all* of them. The system analysis phase and involvement of users will still be critical, especially for ambitious projects. Outsourcing and offshoring can make the feasibility and analysis steps harder by impeding communication.

Try to avoid having additional features and capabilities creep in at each stage (“scope creep” or “feature creep”): decide what you are going to do, how you'll do it, and then “just do it”.

Overall benefits of SDLC as opposed to less structured approaches:

- Easier to estimate time and effort for the project
- Easier to monitor progress
- More control over scope creep
- Can stay closer to budget and deadline
- Easier to integrate work of different contributors
- More communication between users and developers, less disappointment in final results.

Main drawbacks of SDLC:

- Can be cumbersome and slow.
- Can inflate the cost of making small changes or adjustments.

RT also describes some alternatives (or complementary approaches) to SDLC. I don't have experience with them. I do have experience with SDLC, and it is enormously beneficial – especially the formal analysis and design phases.