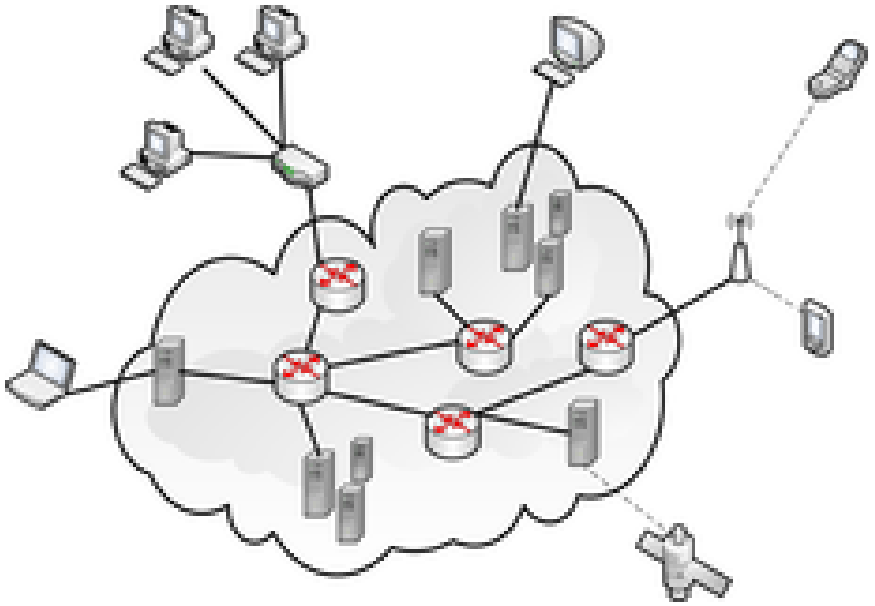
ECE419 - Distributed Systems: Introduction, Motivation & Overview

http://ece419.msrg.utoronto.ca

Slides available in Quercus

(**before** or after lecture)



ADMINISTRATIVE OVERVIEW

419 Online Resources

- Course web site
  - http://ece419.msrg.utoronto.ca
- Quercus
  - Post announcements, assignments, slides
  - Push out (email) notifications
  - Manage marks
- Piazza (best effort service by TAs, only)
  - You should receive an invite or be able to join

Schedule

- Lecture
  - Monday, 4-6 PM
  - Thursday, 5-6 PM
- Tutorial lecture
  - As per announcement, one of the above slots
- Lab
  - Three lab sections, serviced by TAs (**see schedule**!)

ECE 419 in a Nutshell

Lectures (concepts)

Tutorial (project & assignments)

Our Project: **Four Milestones**

Four Assignments

Exams (midterm & final)

Project Overview
(Completed in Teams of 3 Students)

- Course project
  - Build a **distributed storage service**
  - We call it the "***cloud database lab***" as it …
    - **Mimics** conception of storage services at major online players
    - **Conceivably** run in a cloud
    - **But, simplified** for the context of the course
    - Project's "cloudiness" really in the background
- Project Milestones
  - Project divided into **four milestones**
  - We provide **sample code** and **skeleton code**
- Project is to be developed in **Java**
- Optional bonus marks available

Team Rules

- Team must work together on the entire project
- Team selection deadline **January 19th**
- Team selection mechanism available online
- **Do not wait, start to plan, design, implement earlier!**
- Milestone 1 will go live later this week
- May team up with peers in all lab sections
  - **BUT, entire team must be available for team milestone demos in any of its members' scheduled lab sections (a binding agreement)!**

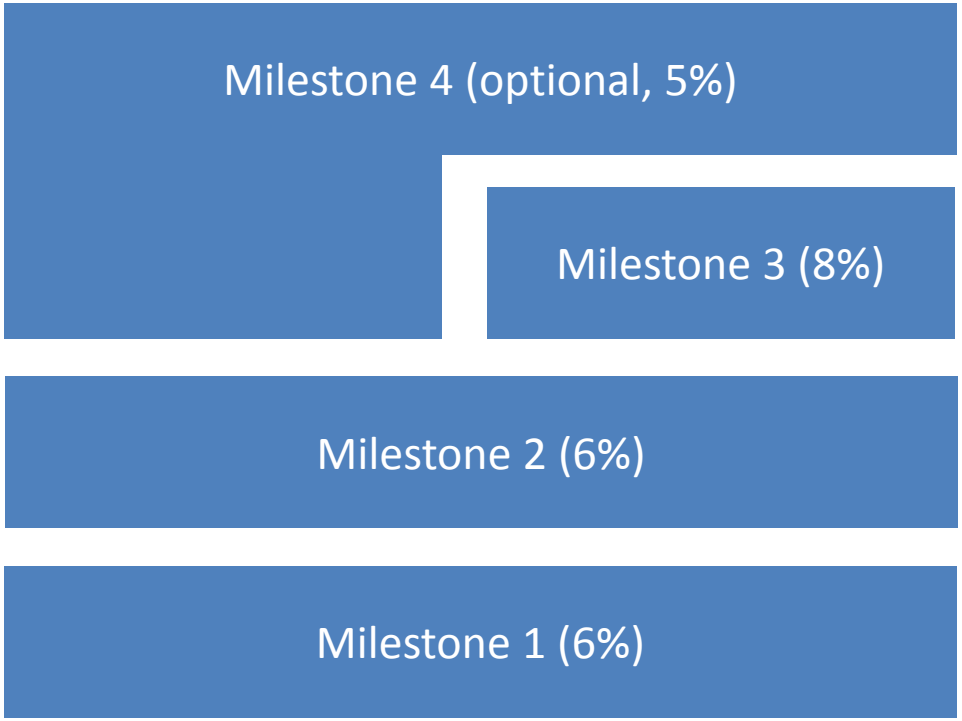
Milestone Due Dates

(Authorative Deadlines see our Website)

Milestone	Due Date
Milestone 1	Sunday, Jan 28th (3 weeks)
Milestone 2	Sunday, Februar 16th (3 weeks)
Milestone 3	Sunday, Mar 22nd (~3 weeks)
Milestone 4 (**Optional at 5%**)	Thursday, April 9th (~2 weeks)

Milestone Dependencies

- M1 to M3 successively build on one another
- M4 builds either on M2 or on M3
- **Highest bonus if M4 builds on M3**



Milestone Evaluation

- Evaluation of submitted artefacts and lab demo of running project (Milestones 1 to 3).
- Submission of **software artefacts** and **demos:** We evaluate submitted artefacts on **unNNN.eecg lab machines**
- Milestone demos during scheduled and slotted lab times
- Only team members attending the demo will receive a mark (demo slot about 15-20 minutes)
- Project and submitted milestones must **run and be buildable on unNNN.eecg lab machines**
  - Linux Debian environment
  - Laptop-based demos are not accepted

Project Management Recommendation

- Designate a project manager for the team
- Do scrum-style, agile development
- Use source control system
  - SVN, git, …, but keep code private
- We'll setup a shared group for your team on the **`unNNN.eecg`** lab machines
- Diligently unit test your code
- **Practice demos**

Assignment Overview
(Completed individually)

- Four assignments throughout semester
- Also, completed by team
- Practice lecture material, **prepare for exams**
- Some assignment solutions discussed in Thursday's lecture by TAs
- **Disclaimer**: Exams may also ask questions relating to course project
- Submitted at **Mailbox XX**, in **SF basement**
- **Assignment 1** will go live later in the week

Assignment Due Dates
(Authorative Deadlines see our Website)

Milestone	Due Date
Assignment 1	Wednesday, Jan. 29th
Assignment 2	Wednesday, Feb. 12th
Assignment 3	Wednesday, Mar 18th
Assignment 4	Wednesday, April 8th

- Overall, 10% of Final Mark
- **Two assignments before midterm**, two after
- Deadline is **noon** on specified date

No Late Course Work Accepted

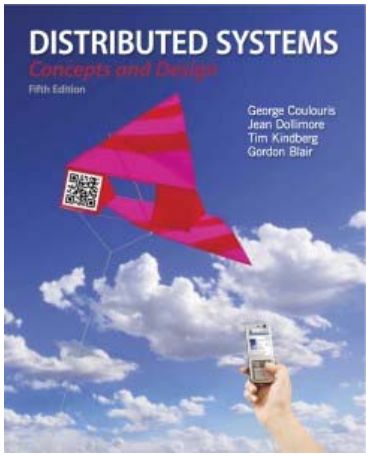
- No late submissions

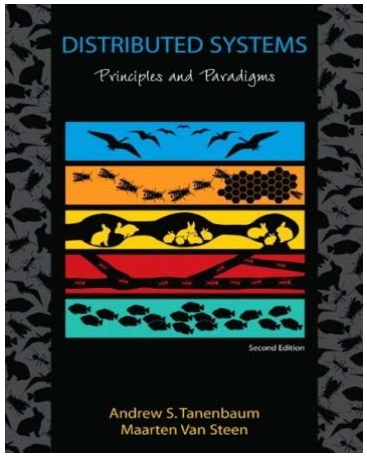
- We will not respond to emails about late submissions

- For legitimate reasons, talk to Head TA or me after lecture (need official documentation)

Reading Material

- **Required reading** posted with each lecture (for the exam)
- **Recommended reading** posted with each lecture (for life)
- Reading draws from selected **overlapping sections** of the below two books and from online resources (roughly 40-50%)
- Other reading from online sources





Example: Week's Reading

- **Required**:
  - Either Chapter 1 TS or Chapter 1 CDKB
  - Fallacies of Distributed Computing Explained By Arnon Rotem-Gal-Oz
- **Recommended**:
  - Deutsch's Fallacies, 10 Years After
  - Introduction to Distributed System Design (Google Code University)

Midterm & Final

- **Both are closed book (Type C*)**
- Cover lectures, labs (project), and assignments
- You are responsible for
  - **All required reading material** assigned
  - **All lecture, project, assignment material**
- Required reading, even if not fully covered in lecture
- **Midterm tentatively set** for week after reading week **(Tue., Feb 28th during lecture time)**
- Midterm about 90 minutes

Midterm & Final

- Allowed Calculator Types
  - **Type 4: No electronic or mechanical computing devices** will be permitted

- Exam Paper Type
  - **Type C:** A "closed book" examination for which the **candidate may prepare, bring to the examination and use, a single aid sheet**, downloaded from the Faculty's website, printed on an 8.5"x11" piece of paper. **Students may enter on both sides of the aid sheet any information they desire**, without restriction, except that nothing may be affixed or appended to it. **Such entries will be handwritten and not mechanically reproduced**.

Mark Breakdown

Component	Mark Portion
Course project	20%
Homework	10%
Midterm	30%
Final	40%

Disclaimer about course project:

- Mark is 10% + 10%
- 10% is a **team mark**
- 10% is by **individual**

Course Scope

What is Not in Scope?

- Middleware abstractions
- Parallel, concurrent programming
- Security (“*The network is secure.*” ☹)
  - One of the 10 fallacies, so watch out!

You May Benefit From Prior Exposure

- Algorithms and data structures
  - Searching, sorting, hash tables, lists, trees, graphs, basic notions of complexity
- Basic programming skills and tools
  - Java, concurrency, multithreading, synchronization
  - Code versioning and concurrent development
- Operating systems concepts
- Computer network and database systems basics

Tentative Course Outline

- **Time** in distributed systems
- **Coordination** and **agreement**
- **Consensus** with Paxos
- **Replication**
- **Consistency** and transactions
- Consistent hashing, **CAP theorem**, web caching
- Distributed file systems (GFS)
- **MapReduce**, Spark
- Peer-to-peer systems, **distributed hash tables (DHTs)**
- **Blockchains**

COURSE PROJECT (FOUR MILESTONES)

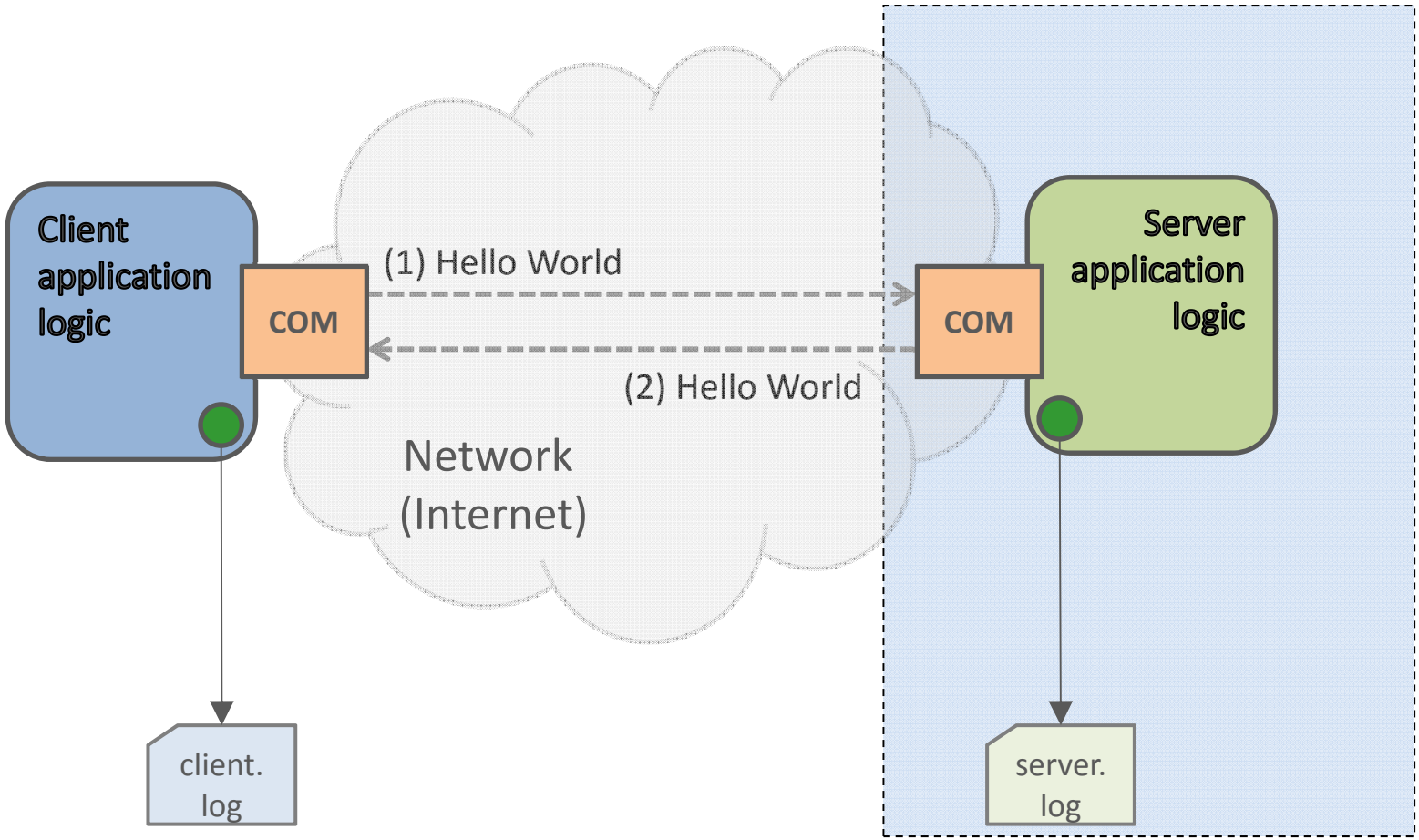
What is the Project About?

- State-of-the-art in NO SQL DBs
  - Google's Big Table
  - Apache Hbase
  - Amazon's Dynamo
  - Facebook's Cassandra
  - Yahoo's PNUTS

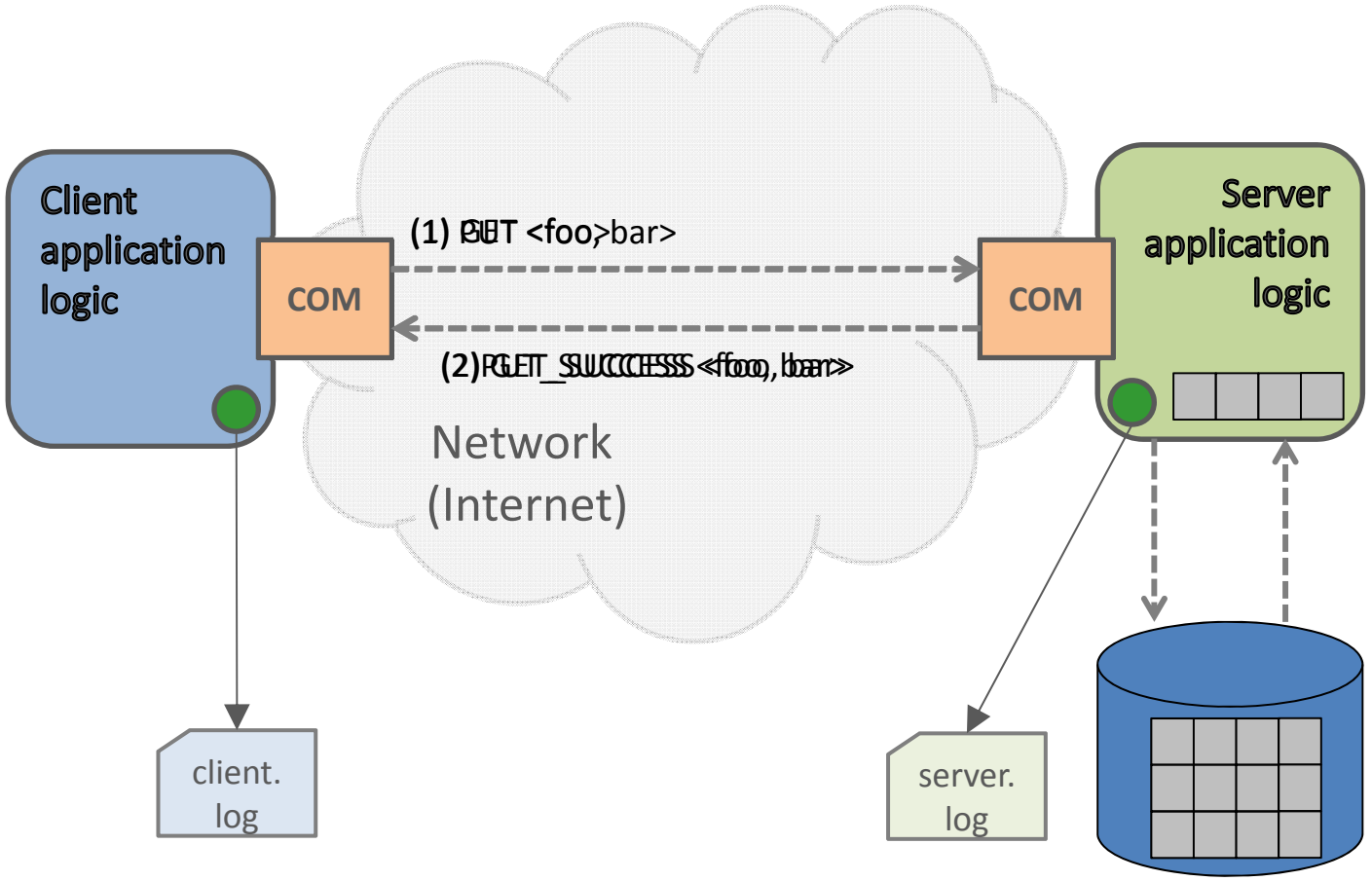
- Build, test, deploy, and evaluate your own cloud database
  - Incrementally design, develop, and evaluate
  - Across four milestones
  - Deliverables: code, reports, demos
  - Exam questions

"*Milestone 0*" – Echo Client/Server

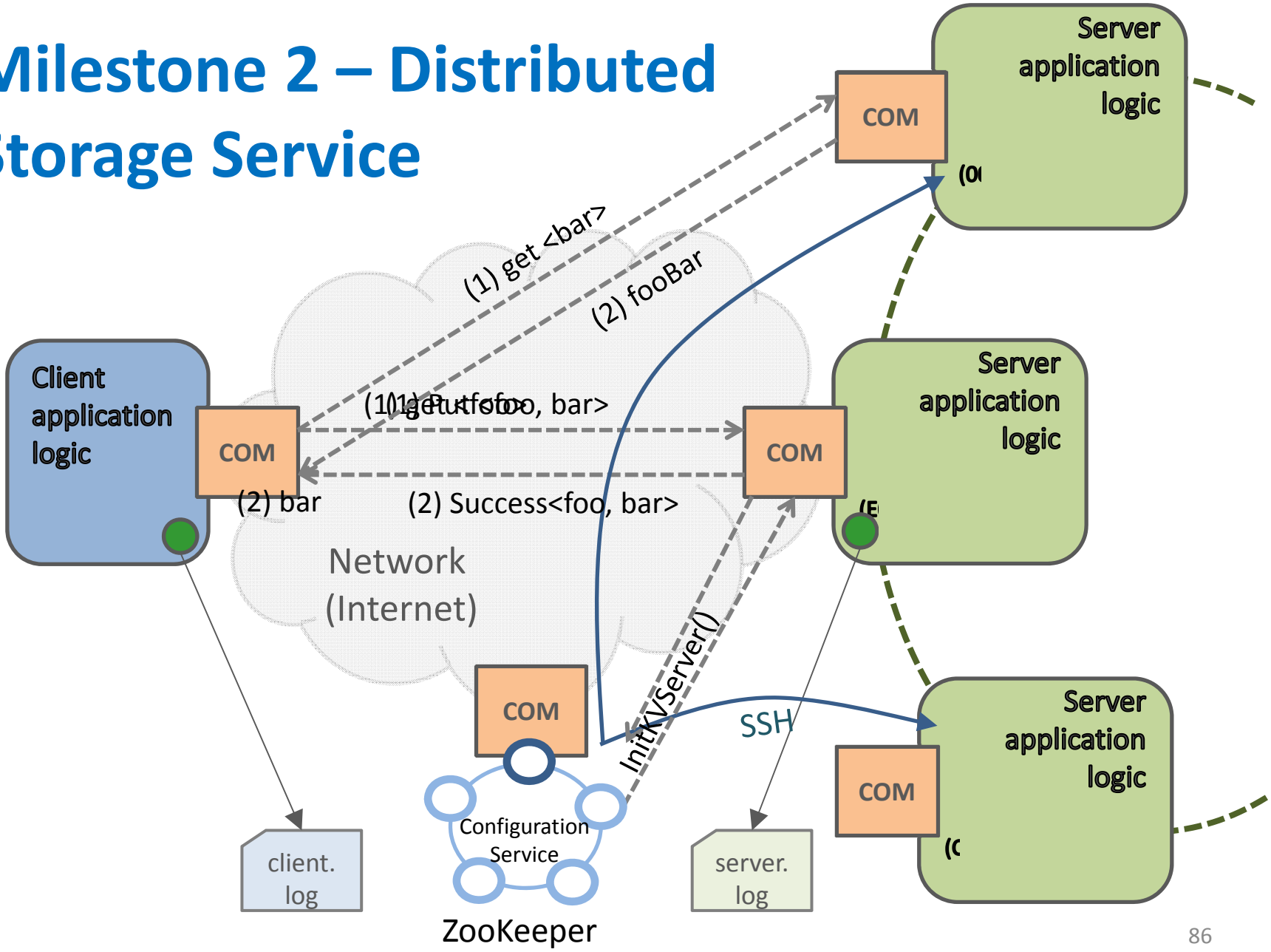
(All Code Provided)



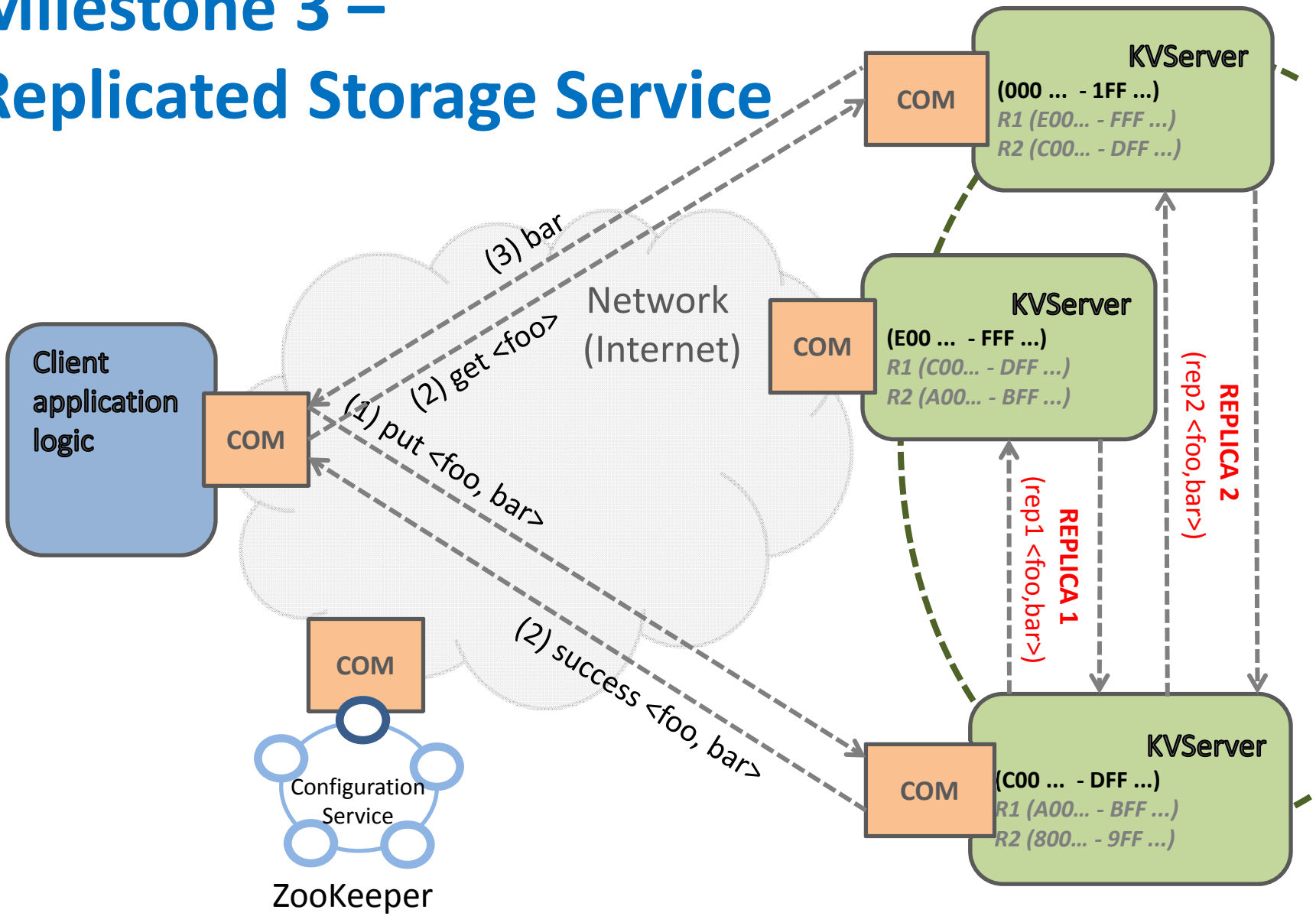
Milestone 1 –
Storage Server & Client



Milestone 2 – Distributed Storage Service

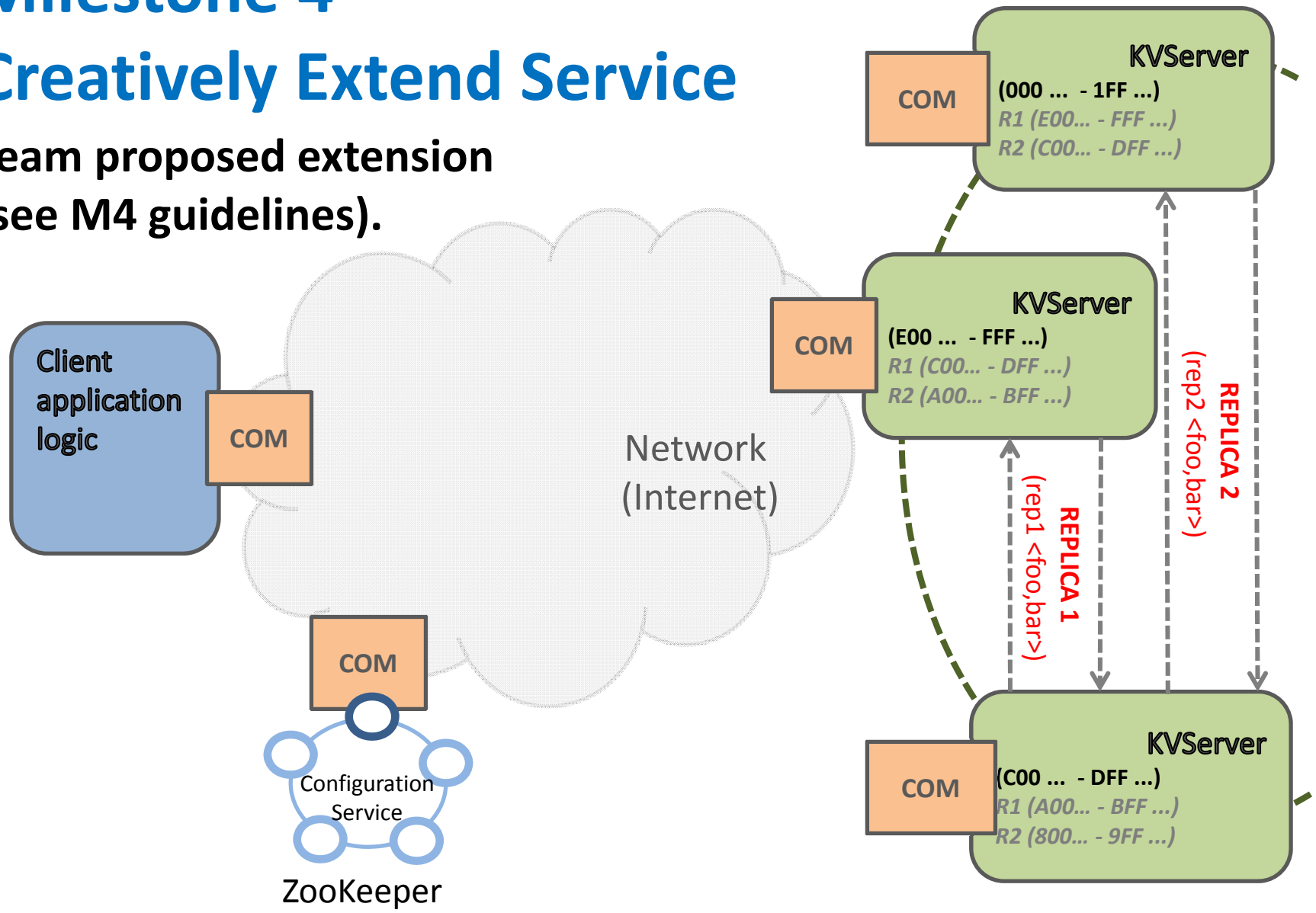


Milestone 3 – Replicated Storage Service



Milestone 4 – Creatively Extend Service

Team proposed extension (see M4 guidelines).



Project Goals & Expectations

- Goals
  - Learn to develop distributed systems
  - Get exposed to the state-of-the-art of today's large-scale storage systems
  - Design, implement, and evaluate your own approach (on a smaller scale)
- Expectations
  - Genuine interest & active participation
  - Bring in your own ideas
  - Deliverables: code, report, demo

Milestone Due Dates

(Authorative Deadlines see our Website)

Milestone	Due Date
Milestone 1	Sunday, Jan. 26th
Milestone 2	Sunday, Feb. 16th
Milestone 3	Sunday, Mar 22nd
Milestone 4	Wednesday, April 8th

- Overall, 10% of Final Mark
- **Two milestones before midterm**, two after

Learning Objectives 419

- Design and develop **architectures for distributed systems** and applications
- Apply **foundational principles** in the development of distributed systems
- Understand **properties of common building blocks** applicable to distributed systems design
- Understand the **complexities** involved in developing a distributed system (e.g., machine and network failures, concurrency, etc.)
- Inspire **system-driven** research with **implementation and evaluation**.

FALLACIES

Distributed Systems Design Fallacies

- **Assumptions** (novice) designers of distributed systems often make **that turn out to be false**
- Originated in 1994 by Peter Deutsch, Sun Fellow, Sun Microsystems

- **The 8 fallacies**
  - The network is reliable.
  - Latency is zero.
  - Bandwidth is infinite.
  - The network is secure.
  - Topology doesn't change.
  - There is one administrator.
  - Transport cost is zero.
  - The network is homogeneous

Tentative Course Outline
(Topic Order May Differ)

- Distributed systems introduction
- **Network programming** basics
- **Time** in distributed systems
- **Consensus** and **agreement**
- **Consistent hashing** and caching
- **Consistency** and **replication**
- Replicated state machines, **coordination**, failure detectors
- MapReduce
- **Peer-to-peer** principles, systems, and applications

The network is reliable

- Switches & routers rarely fail
  - Mean time between failures is very high (years!)
- *Why then is this a fallacy?*
  - Power supply, hard disk, node failures
  - Incorrect configurations
  - Bugs, dependency on external services

- **Effect** is that **application** hangs or **crashes**

The network is reliable: Implications for design

- Redundancy
  - Infrastructure & hardware
  - Software systems, middleware & application

- Catch exceptions, check error codes, react accordingly
- Prepare to retry connecting upon timeouts
- Acknowledge, reply or use negative acknowledgements
- Identify & ignore duplicates
- Use idempotent operations (out of order)
- Verify message integrity

Bottom line:

The network is unreliable and the designer needs to deal with it!

Latency is zero

- *"Making a call over the wire is like making a local call."*
- Informally speaking
  - Latency is the time it takes for data to move from one place to another
  - Bandwidth is how much data can be transferred during that time (bits per second)
- Latency is capped by the maximum speed of information transmission, i.e., the speed of light
  - at ~300,000 kilometres per second, round trip time between US-Europe (~ 8K km) is ~60ms
- Bandwidth (& its use) keeps on growing

Latency is zero vs. cost of a method call

- **Local call** is essentially a *Push* and a *Jump to Subroutine*
- **System call** is taken care of by OS (100s of assembly instructions)
- Call across a LAN involves **system calls** on **caller** and **callee** and network latency
- Call across a WAN … transmission delays etc.
- Strive to make **as few calls as possible**, **moving as much data as possible**
- Trading off computing for data transmitted (cf. "bandwidth is not infinite" & I/O is expensive).

The network is secure, ☹

sshd:
  Invalid Users:
    Unknown Account: 2241 Time(s)
  Authentication Failures:
    dan (<IP Address>): 5 Time(s)
    postfix (<IP Address>): 1 Time(s)
    mysql (<hostname>): 1 Time(s)
    postgres (<IP Address>): 3 Time(s)
    root (g<hostname>): 8 Time(s)
    postgres (<hostname>): 3 Time(s)
    smmsp (<IP Address>): 1 Time(s)
    david (<hostname>): 2 Time(s)
    postfix (<hostname>): 1 Time(s)
    david (<IP Address>): 3 Time(s)
    dan (<hostname>): 5 Time(s)
    smmsp (<hostname>): 1 Time(s)
    uucp (<hostname>): 1 Time(s)

Failed logins from these:
  aaron/password from 211.43.206.53: 1 Time(s)
  abdullah/password from 211.43.206.53: 1 Time(s)
  abraham/password from 211.43.206.53: 1 Time(s)
  abram/password from 211.43.206.53: 1 Time(s)
  account/password from 142.150.237.133: 1 Time(s)
  account/password from 211.43.206.53: 1 Time(s)
  adam/password from 211.43.206.53: 3 Time(s)
  addison/password from 211.43.206.53: 1 Time(s)
  aditya/password from 211.43.206.53: 1 Time(s)
  admin/password from 142.150.237.133: 18 Time(s)
  admin/password from 211.43.206.53: 18 Time(s)
  administrator/password from 142.150.237.133: 3 Time(s)
  administrator/password from 211.43.206.53: 3 Time(s)
  adolfo/password from 211.43.206.53: 1 Time(s)
  adrianna/password from 142.150.237.133: 1 Time(s)

Summary: The 8 fallacies

- **The network is reliable**.
- **Latency is zero**.
- Bandwidth is infinite.
- **The network is secure.**
- Topology doesn't change.
- There is one administrator.
- Transport cost is zero.
- The network is homogeneous.