DIRAS: Distributed Image Reconstruction in Adversarial Scenario

Under the supervision of Prof. Sanand Dilip Amita Athalye

Shailesh Mishra - 17EE35014

Department of Electrical Engineering, IIT Kharagpur

November 25, 2021

Outline

Motivation

Motivation - CPS

- **CyberPhysical Systems (CPSs)** - Integrate features physical processes with computational processes and communication networks [Baheti and Gill, 2011]

Motivation - CPS

- **CyberPhysical Systems (CPSs)** - Integrate features physical processes with computational processes and communication networks [Baheti and Gill, 2011]
- Sister Technology - **Internet of Things** - connecting all the devices over the internet for improved processing [Atzori et al., 2010]

Motivation - CPS

- **CyberPhysical Systems (CPSs)** - Integrate features physical processes with computational processes and communication networks [Baheti and Gill, 2011]
- Sister Technology - **Internet of Things** - connecting all the devices over the internet for improved processing [Atzori et al., 2010]
- CPS and IoT together - have revolutionized the functioning of various systems

Motivation - CPS

- **CyberPhysical Systems (CPSs)** - Integrate features physical processes with computational processes and communication networks [Baheti and Gill, 2011]
- Sister Technology - **Internet of Things** - connecting all the devices over the internet for improved processing [Atzori et al., 2010]
- CPS and IoT together - have revolutionized the functioning of various systems
- Reasons for success of CPS and IoT:

Motivation - CPS

- **CyberPhysical Systems (CPSs)** - Integrate features physical processes with computational processes and communication networks [Baheti and Gill, 2011]
- Sister Technology - **Internet of Things** - connecting all the devices over the internet for improved processing [Atzori et al., 2010]
- CPS and IoT together - have revolutionized the functioning of various systems
- Reasons for success of CPS and IoT:
  - Ability to acquire a large amount of data from various sources

Motivation - CPS

- **CyberPhysical Systems (CPSs)** - Integrate features physical processes with computational processes and communication networks [Baheti and Gill, 2011]
- Sister Technology - **Internet of Things** - connecting all the devices over the internet for improved processing [Atzori et al., 2010]
- CPS and IoT together - have revolutionized the functioning of various systems
- Reasons for success of CPS and IoT:
  - Ability to acquire a large amount of data from various sources
  - Huge computational power of all the devices as a whole

Motivation - CPS

- **CyberPhysical Systems (CPSs)** - Integrate features physical processes with computational processes and communication networks [Baheti and Gill, 2011]
- Sister Technology - **Internet of Things** - connecting all the devices over the internet for improved processing [Atzori et al., 2010]
- CPS and IoT together - have revolutionized the functioning of various systems
- Reasons for success of CPS and IoT:
  - Ability to acquire a large amount of data from various sources
  - Huge computational power of all the devices as a whole
- Wide integration of CPS and IoT

Motivation - Image

- CPS and IoT process various kinds of data - image, audio, temperature, etc.

Motivation - Image

- CPS and IoT process various kinds of data - image, audio, temperature, etc.
- Image - One of the most prevalent

Motivation - Image

- CPS and IoT process various kinds of data - image, audio, temperature, etc.
- Image - One of the most prevalent
- Analysis of image plays a key role in agriculture [Vibhute and Bodhe, 2012], medical systems [Dougherty, 2009], remote sensing [Chen, 2012], robotics [Kurka and Salazar, 2019]

Motivation - Image

- CPS and IoT process various kinds of data - image, audio, temperature, etc.
- Image - One of the most prevalent
- Analysis of image plays a key role in agriculture [Vibhute and Bodhe, 2012], medical systems [Dougherty, 2009], remote sensing [Chen, 2012], robotics [Kurka and Salazar, 2019]
- Image data holds great value in numerous fields

Motivation - Problem

- All these new advancements have brought in a new issue - **security of data**

Motivation - Problem

- All these new advancements have brought in a new issue - **security of data**
- Huge amount of image data that becomes difficult to manage

Motivation - Problem

- All these new advancements have brought in a new issue - **security of data**
- Huge amount of image data that becomes difficult to manage
- Conventional architecture of CPS - centralized (client-server based)

Motivation - Problem

- All these new advancements have brought in a new issue - **security of data**
- Huge amount of image data that becomes difficult to manage
- Conventional architecture of CPS - centralized (client-server based)
  - Single point of failure

Motivation - Problem

- All these new advancements have brought in a new issue - **security of data**
- Huge amount of image data that becomes difficult to manage
- Conventional architecture of CPS - centralized (client-server based)
  - Single point of failure
  - Not scalable

Motivation - Problem

- All these new advancements have brought in a new issue - **security of data**
- Huge amount of image data that becomes difficult to manage
- Conventional architecture of CPS - centralized (client-server based)
  - Single point of failure
  - Not scalable
- Images acquired - hold great value

Motivation - Problem

- All these new advancements have brought in a new issue - **security of data**
- Huge amount of image data that becomes difficult to manage
- Conventional architecture of CPS - centralized (client-server based)
  - Single point of failure
  - Not scalable
- Images acquired - hold great value
- If those images are tampered with - wrong processing that can prove to be fatal at times

Motivation - Problem

- All these new advancements have brought in a new issue - **security of data**
- Huge amount of image data that becomes difficult to manage
- Conventional architecture of CPS - centralized (client-server based)
  - Single point of failure
  - Not scalable
- Images acquired - hold great value
- If those images are tampered with - wrong processing that can prove to be fatal at times
- Such large amount of data has been exploited by adversaries to execute attacks such as MITM attacks [Rong-xiao et al., 2020], DoS attacks [Liang et al., 2016] and dropping of packets

Motivation - Problem

- All these new advancements have brought in a new issue - **security of data**
- Huge amount of image data that becomes difficult to manage
- Conventional architecture of CPS - centralized (client-server based)
  - Single point of failure
  - Not scalable
- Images acquired - hold great value
- If those images are tampered with - wrong processing that can prove to be fatal at times
- Such large amount of data has been exploited by adversaries to execute attacks such as MITM attacks [Rong-xiao et al., 2020], DoS attacks [Liang et al., 2016] and dropping of packets
- **These issues need to be addressed**

Motivation - Solution

- We propose **DIRAS: Distributed Image Reconstruction in Adversarial Scenarios**

Motivation - Solution

- We propose **DIRAS: Distributed Image Reconstruction in Adversarial Scenarios**
- DIRAS is distributed which makes it fault-tolerant

Motivation - Solution

- We propose **DIRAS: Distributed Image Reconstruction in Adversarial Scenarios**
- DIRAS is distributed which makes it fault-tolerant
- Deploys data splitting for improving privacy and efficiency of data sharing

Motivation - Solution

- We propose **DIRAS: Distributed Image Reconstruction in Adversarial Scenarios**
- DIRAS is distributed which makes it fault-tolerant
- Deploys data splitting for improving privacy and efficiency of data sharing
- Uses **Robust Principal Component Analysis** and **Matrix Completion** for cleaning the tampered image

Motivation - Solution

- We propose **DIRAS: Distributed Image Reconstruction in Adversarial Scenarios**
- DIRAS is distributed which makes it fault-tolerant
- Uses data splitting for improving privacy and efficiency of data sharing
- Uses Robust Principal Component Analysis and Matrix Completion for cleaning the tampered image
- Incorporates other mechanisms to mitigate the effect of other security attacks

Literature Review

Literature Review

- **Control-theoretic Methods** [Pasqualetti et al., 2015]

Literature Review

- **Control-theoretic Methods** [Pasqualetti et al., 2015]
- **Optimal Linear Cyber Attack** [Guo et al., 2016]

Literature Review

- **Control-theoretic Methods** [Pasqualetti et al., 2015]
- **Optimal Linear Cyber Attack** [Guo et al., 2016]
- **Distributed optimization** [Nedić and Liu, 2018]

Literature Review

- **Control-theoretic Methods** [Pasqualetti et al., 2015]
- **Optimal Linear Cyber Attack** [Guo et al., 2016]
- **Distributed optimization** [Nedić and Liu, 2018]
- **Paxos** [Lamport et al., 2001]

Literature Review

- **Control-theoretic Methods** [Pasqualetti et al., 2015]
- **Optimal Linear Cyber Attack** [Guo et al., 2016]
- **Distributed optimization** [Nedić and Liu, 2018]
- **Paxos** [Lamport et al., 2001]
- **Blockchain** [Nakamoto, 2008]

Literature Review

- **Control-theoretic Methods** [Pasqualetti et al., 2015]
- **Optimal Linear Cyber Attack** [Guo et al., 2016]
- **Distributed optimization** [Nedić and Liu, 2018]
- **Paxos** [Lamport et al., 2001]
- **Blockchain** [Nakamoto, 2008]
- **Image Reconstruction** [Belthangady and Royer, 2019, Wang et al., 2020]

Literature Review

- **Control-theoretic Methods** [Pasqualetti et al., 2015]
- **Optimal Linear Cyber Attack** [Guo et al., 2016]
- **Distributed optimization** [Nedić and Liu, 2018]
- **Paxos** [Lamport et al., 2001]
- **Blockchain** [Nakamoto, 2008]
- **Image Reconstruction** [Belthangady and Royer, 2019, Wang et al., 2020]

DIRAS combines the concept of security in CPS, distributed optimization and image reconstruction for secure and scalable reconstruction of images.

Background

Background - Robust Principal Component Analysis (RPCA)

- RPCA is developed to make Principal Component Analysis (PCA) robust [Candès et al., 2011]

Background - Robust Principal Component Analysis (RPCA)

- RPCA is developed to make Principal Component Analysis (PCA) robust [Candès et al., 2011]
- About PCA

Background - Robust Principal Component Analysis (RPCA)

- RPCA is developed to make Principal Component Analysis (PCA) robust [Candès et al., 2011]
- About PCA
  - Data matrix - Sum of a low-rank matrix and a sparse noise

Background - Robust Principal Component Analysis (RPCA)

- RPCA is developed to make Principal Component Analysis (PCA) robust [Candès et al., 2011]
- About PCA
  - Data matrix - Sum of a low-rank matrix and a sparse noise
  - Goal - given the data matrix, can we obtain the low-rank matrix and the sparse matrix?

Background - Robust Principal Component Analysis (RPCA)

- RPCA is developed to make Principal Component Analysis (PCA) robust [Candès et al., 2011]
- About PCA
  - Data matrix - Sum of a low-rank matrix and a sparse noise
  - Goal - given the data matrix, can we obtain the low-rank matrix and the sparse matrix?
  - PCA helps to achieve this goal

Background - Robust Principal Component Analysis (RPCA)

- RPCA is developed to make Principal Component Analysis (PCA) robust [Candès et al., 2011]
- About PCA
  - Data matrix - Sum of a low-rank matrix and a sparse noise
  - Goal - given the data matrix, can we obtain the low-rank matrix and the sparse matrix?
  - PCA helps to achieve this goal
  - PCA - based on a convex optimization problem called Principal Component Pursuit

Background - Robust Principal Component Analysis (RPCA)

- RPCA is developed to make Principal Component Analysis (PCA) robust [Candès et al., 2011]
- About PCA
  - Data matrix - Sum of a low-rank matrix and a sparse noise
  - Goal - given the data matrix, can we obtain the low-rank matrix and the sparse matrix?
  - PCA helps to achieve this goal
  - PCA - based on a convex optimization problem called Principal Component Pursuit
- However, in case of gross errors - performance of PCA degrades

Background - Robust Principal Component Analysis (RPCA)

- RPCA is developed to make Principal Component Analysis (PCA) robust [Candès et al., 2011]
- About PCA
  - Data matrix - Sum of a low-rank matrix and a sparse noise
  - Goal - given the data matrix, can we obtain the low-rank matrix and the sparse matrix?
  - PCA helps to achieve this goal
  - PCA - based on a convex optimization problem called Principal Component Pursuit
- However, in case of gross errors - performance of PCA degrades
- Need for making PCA robust

Background - Robust Principal Component Analysis (RPCA)

- RPCA is developed to make Principal Component Analysis (PCA) robust [Candès et al., 2011]
- About PCA
  - Data matrix - Sum of a low-rank matrix and a sparse noise
  - Goal - given the data matrix, can we obtain the low-rank matrix and the sparse matrix?
  - PCA helps to achieve this goal
  - PCA - based on a convex optimization problem called Principal Component Pursuit
- However, in case of gross errors - performance of PCA degrades
- Need for making PCA robust
- RPCA helps in extracting the low-rank matrix and the sparse matrix from the data matrix in case of gross errors

Background - Matrix Completion

- Matrix Completion [Keshavan et al., 2010] will be used when RPCA won't be capable to reconstruct the image

Background - Matrix Completion

- Matrix Completion [Keshavan et al., 2010] will be used when RPCA won't be capable to reconstruct the image
- Occurs when the node reconstructing the image does not receive chunks of the image - parts of an image are missing

Background - Matrix Completion

- Matrix Completion [Keshavan et al., 2010] will be used when RPCA won't be capable to reconstruct the image
- Occurs when the node reconstructing the image does not receive chunks of the image - parts of an image are missing
- Matrix Completion is used to obtain those parts of the images that have been lost by the framework

System Design

System Design - Assumptions

There are a few assumptions that have been made while designing DIRAS

System Design - Assumptions

There are a few assumptions that have been made while designing DIRAS

- Monitor nodes (the nodes that process and reconstruct images) are connected over a peer-to-peer network, i.e., every monitor node is connected to another

System Design - Assumptions

There are a few assumptions that have been made while designing DIRAS

- Monitor nodes (the nodes that process and reconstruct images) are connected over a peer-to-peer network, i.e., every monitor node is connected to another
- Monitor nodes are trusted and do not act maliciously

System Design - Assumptions

There are a few assumptions that have been made while designing DIRAS

- Monitor nodes (the nodes that process and reconstruct images) are connected over a peer-to-peer network, i.e., every monitor node is connected to another
- Monitor nodes are trusted and do not act maliciously
- Network is synchronous

System Design - Assumptions

There are a few assumptions that have been made while designing DIRAS

- Monitor nodes (the nodes that process and reconstruct images) are connected over a peer-to-peer network, i.e., every monitor node is connected to another
- Monitor nodes are trusted and do not act maliciously
- Network is synchronous
- Monitors have enough computation power to run RPCA and matrix completion algorithms

System Design - Assumptions

There are a few assumptions that have been made while designing DIRAS

- Monitor nodes (the nodes that process and reconstruct images) are connected over a peer-to-peer network, i.e., every monitor node is connected to another
- Monitor nodes are trusted and do not act maliciously
- Network is synchronous
- Monitors have enough computation power to run RPCA and matrix completion algorithms
- Sensors have enough computation power to split images into chunks

System Design - System Architecture

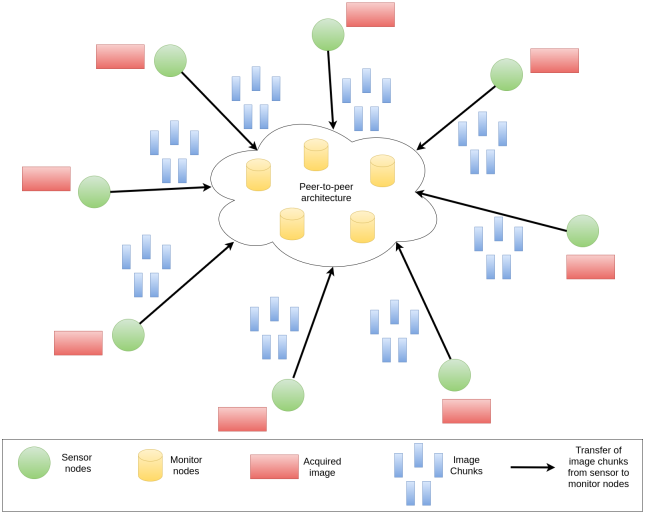

Figure: High level architecture of DIRAS

System Design - System Architecture

- Sensor nodes
- Monitor nodes
- Peer-to-peer network

System Design - System Architecture

- **Sensor nodes**
- Monitor nodes
- Peer-to-peer network

System Design - System Architecture

- **Sensor nodes**
  - Acquire images from surroundings
- Monitor nodes
- Peer-to-peer network

System Design - System Architecture

- **Sensor nodes**
  - Acquire images from surroundings
  - Send the images to the monitor nodes for further analysis
- Monitor nodes
- Peer-to-peer network

System Design - System Architecture

- Sensor nodes
- **Monitor nodes**
- Peer-to-peer network

System Design - System Architecture

- Sensor nodes
- **Monitor nodes**
  - Obtain images from the sensors
- Peer-to-peer network

System Design - System Architecture

- Sensor nodes
- **Monitor nodes**
  - Obtain images from the sensors
  - Reconstruct the whole image which is used for further analysis
- Peer-to-peer network

System Design - System Architecture

- Sensor nodes
- **Monitor nodes**
  - Obtain images from the sensors
  - Reconstruct the whole image which is used for further analysis
  - Remove the noise added to the image by an attacker
- Peer-to-peer network

System Design - System Architecture

- Sensor nodes
- Monitor nodes
- **Peer-to-peer network**

System Design - System Architecture

- Sensor nodes
- Monitor nodes
- **Peer-to-peer network**
  - Monitors are connected over a peer-to-peer network (p2p network)

System Design - System Architecture

- Sensor nodes
- Monitor nodes
- **Peer-to-peer network**
  - Monitors are connected over a peer-to-peer network (p2p network)
  - Ensures that a monitor can share packet with any other monitor

System Design - System Architecture

- Sensor nodes
- Monitor nodes
- **Peer-to-peer network**
  - Monitors are connected over a peer-to-peer network (p2p network)
  - Ensures that a monitor can share packet with any other monitor
  - Reasonable assumption considering the advent of IoT

System Design - Functionality

- Monitor Position Assignment
- Image acquisition and splitting
- Lightweight Leader Selection Algorithm
- Image Reconstruction
- Improvement in DIRAS: Load Balancing

We explain each of these steps in the next few slides.

System Design - Functionality

- **Monitor Position Assignment** - Occurs every epoch time (Δ)
- Image acquisition and splitting
- Lightweight Leader Selection Algorithm
- Image Reconstruction
- Improvement in DIRAS: Load Balancing

System Design - Functionality

Monitor Position Assignment

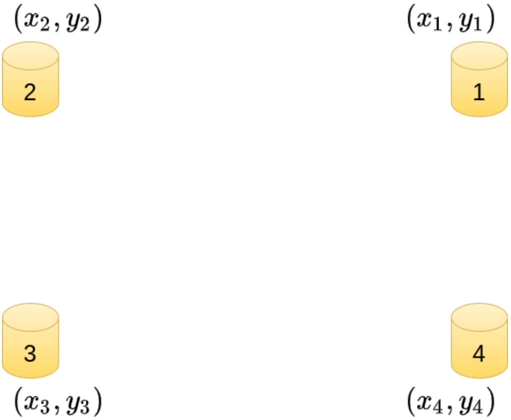

Figure: Monitors generate (pseudo)random coordinates

System Design - Functionality

Monitor Position Assignment

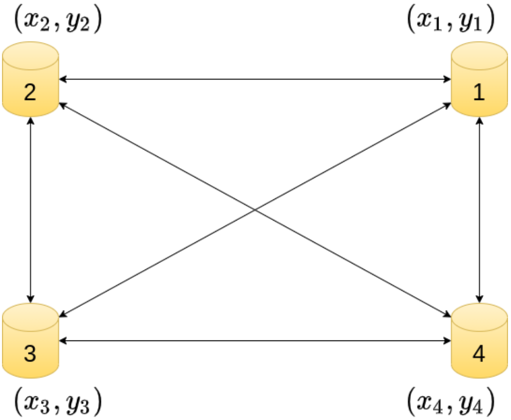

Figure: Monitors broadcast the random coordinates to each other

System Design - Functionality

Monitor Position Assignment

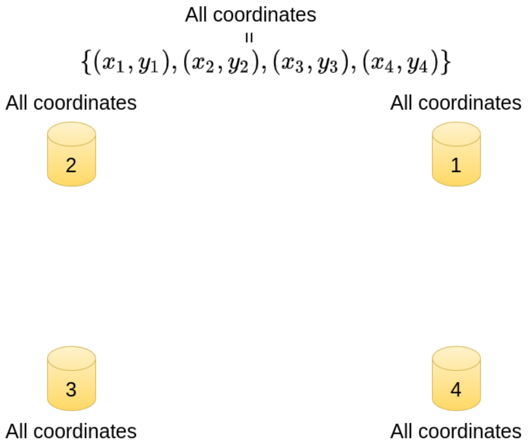

Figure: Monitors acquire and store the coordinates of all other monitors

System Design - Functionality

- Monitor Position Assignment
- **Image acquisition and splitting**
- Lightweight Leader Selection Algorithm
- Image Reconstruction
- Improvement in DIRAS: Load Balancing

System Design - Functionality

Image acquisition and splitting

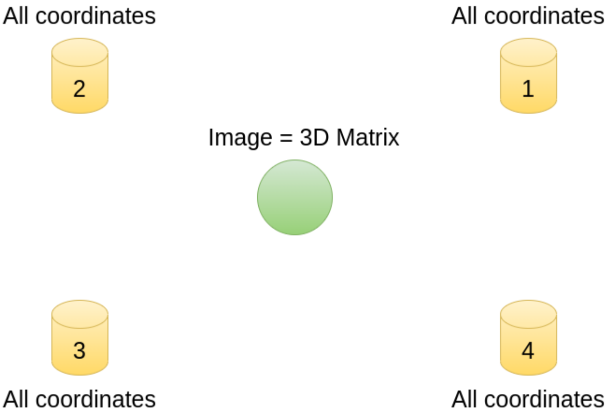

Figure: Sensor acquires image

System Design - Functionality

Image acquisition and splitting

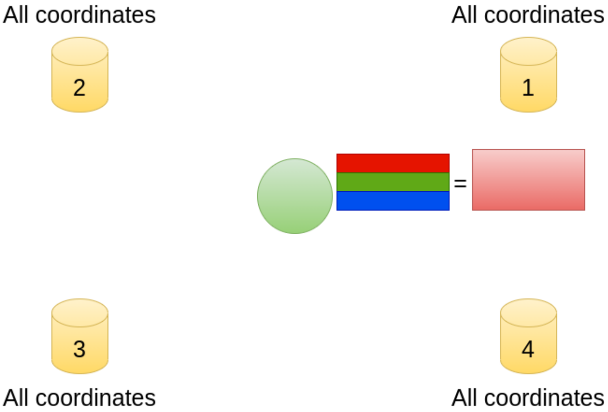

Figure: Sensor stacks the R, G and B components vertically to create a 2-D matrix

System Design - Functionality

Image acquisition and splitting

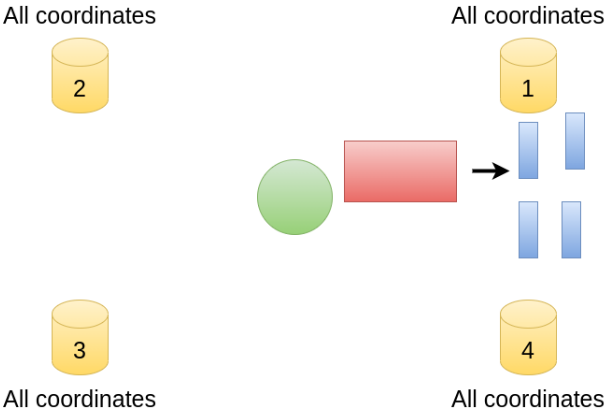

Figure: Sensor splits the image into chunks

System Design - Functionality

Image acquisition and splitting

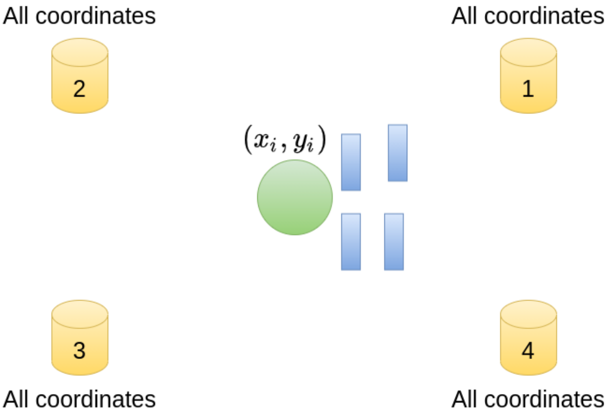

Figure: Sensor generates a random coordinate

System Design - Functionality

Image acquisition and splitting

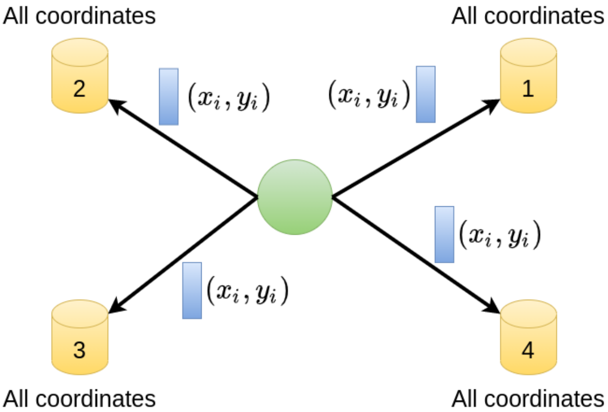

Figure: Sensor sends coordinates and chunks to the monitors

System Design - Functionality

- Monitor Position Assignment
- Image acquisition and splitting
- **Lightweight Leader Selection Algorithm**
- Image Reconstruction
- Improvement in DIRAS: Load Balancing

System Design - Functionality

Lightweight Leader Selection Algorithm

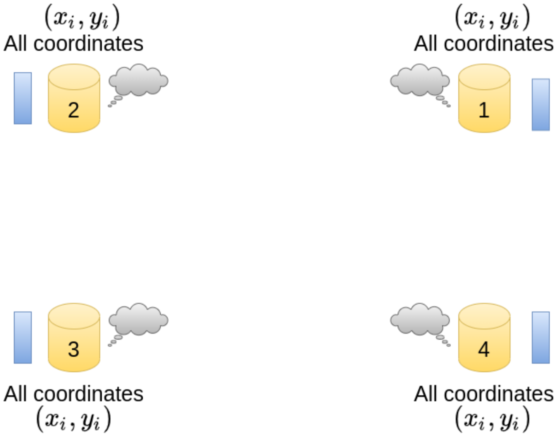

Figure: Monitors find the distance between (x_i, y_i) and all (x_j, y_j) using $D_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$, where j = $\{1, 2, 3, 4\}$

System Design - Functionality

Lightweight Leader Selection Algorithm

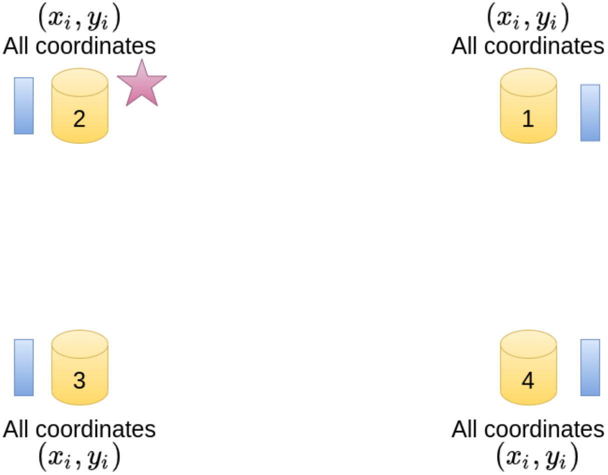

Figure: Monitor with the shortest distance from (x_i, y_i) is the leader

System Design - Functionality

- Monitor Position Assignment
- Image acquisition and splitting
- Lightweight Leader Selection Algorithm
- **Image Reconstruction**
- Improvement in DIRAS: Load Balancing

System Design - Functionality

Image Reconstruction

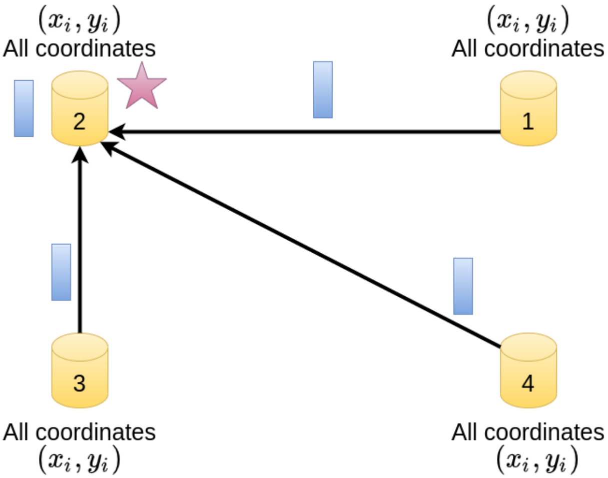

Figure: Monitors send chunks of image to the leader

System Design - Functionality

Image Reconstruction

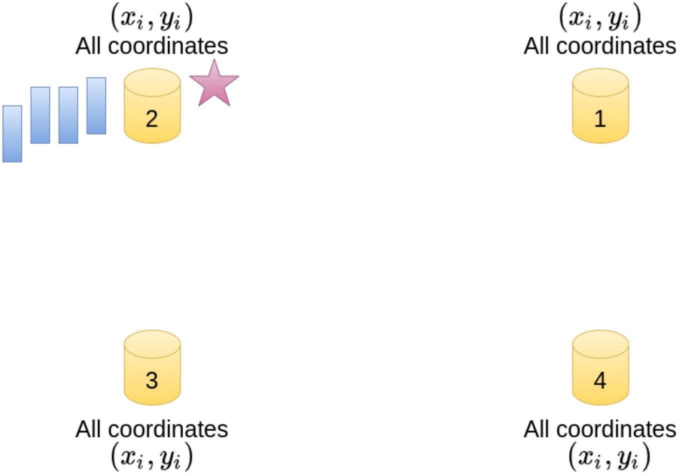

Figure: Leader gets all the chunks from the monitors

System Design - Functionality

Image Reconstruction

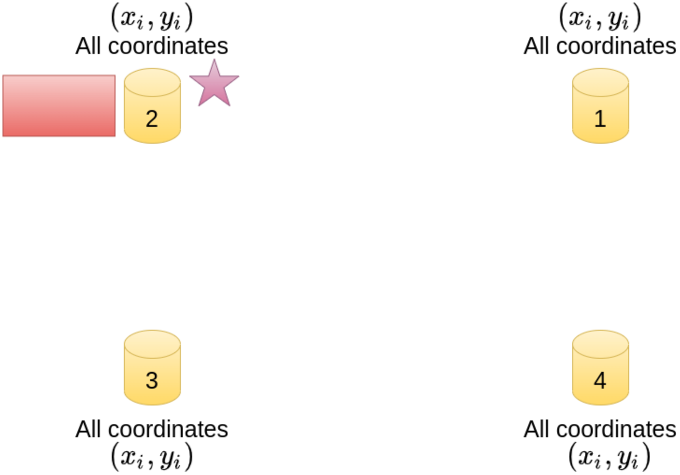

Figure: Leader combines all the chunks to form a single 2D matrix

System Design - Functionality

Image Reconstruction

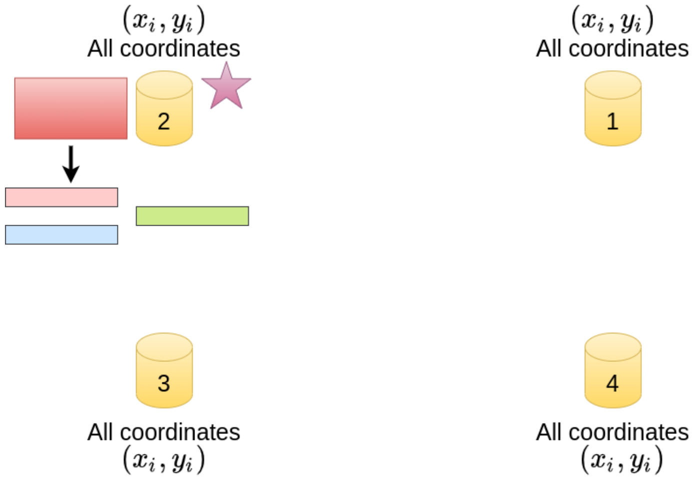

Figure: Leader separates the R, G, and B components of the image

System Design - Functionality

Image Reconstruction

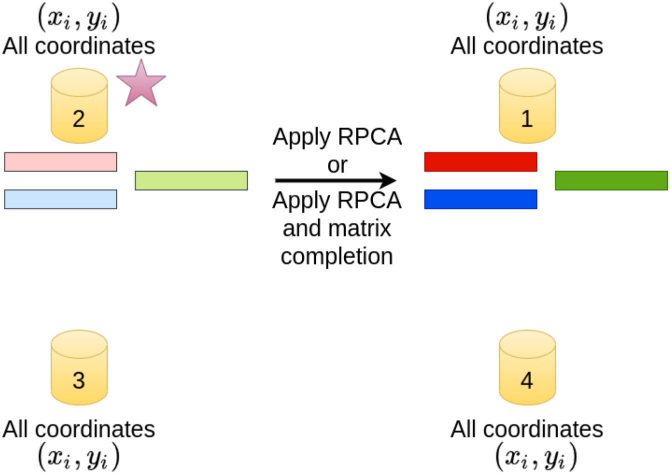

Figure: Leader applies RPCA or RPCA along with matrix completion to remove noise and reconstruct the matrix

System Design - Functionality

Image Reconstruction

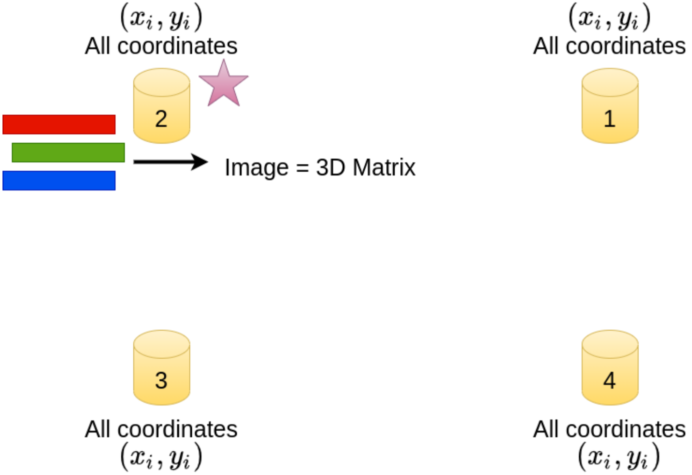

Figure: Leader combines the R, G, and B components of the image to obtain the entire image

System Design - Functionality

When and why do we need RPCA?

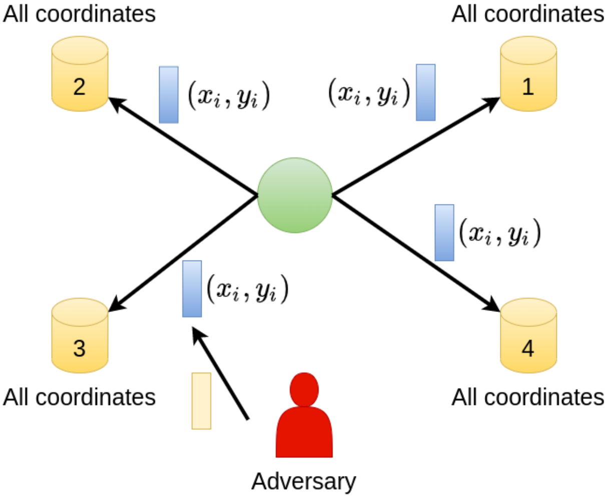

Figure: Attacker adds injects false data to the chunk

System Design - Functionality

When and why do we need RPCA with matrix completion?

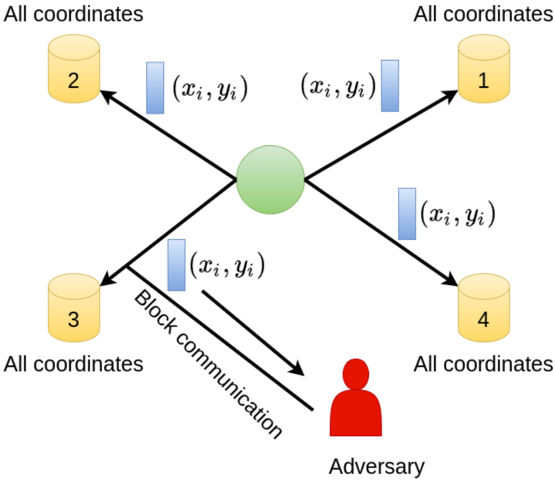

Figure: Attacker adds injects false data to the chunk

System Design - Functionality

- Monitor Position Assignment
- Image acquisition and splitting
- Lightweight Leader Selection Algorithm
- Image Reconstruction
- **Improvement in DIRAS: Load Balancing**

System Design - Functionality

Improvement in DIRAS: Load Balancing

- Leader selection algorithm in DIRAS:

$$L(I_i) = j, \quad \text{s.t.} \quad \min_{j} \quad D_{ij}$$

$$\text{where} \quad D_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}, \quad \text{j} = 1, ..., \mathbf{MN}$$

System Design - Functionality

Improvement in DIRAS: Load Balancing

- Leader selection algorithm in DIRAS:

$$Leader = j, \quad \text{s.t.} \quad \min_{j} \quad D_{ij}$$

$$\text{where} \quad D_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}, \quad \mathrm{j} = 1, ..., \mathbf{MN}$$

- x_i, y_i, x_j, y_j are all random numbers

System Design - Functionality

Improvement in DIRAS: Load Balancing

- Leader selection algorithm in DIRAS:

$$\mathit{Leader} = j, \quad \text{s.t.} \quad \min_{j} \quad D_{ij}$$

$$\text{where} \quad D_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}, \quad \text{j} = 1, ..., \mathbf{MN}$$

- x_i, y_i, x_j, y_j are all random numbers
- Therefore, some monitors will have to reconstruct more images, while some will have to regenerate less images

System Design - Functionality

Improvement in DIRAS: Load Balancing

- Leader selection algorithm in DIRAS:

$$Leader = j, \quad \text{s.t.} \quad \min_{j} \quad D_{ij}$$

$$\text{where} \quad D_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}, \quad \mathrm{j} = 1, ..., \mathbf{MN}$$

- x_i, y_i, x_j, y_j are all random numbers
- Therefore, some monitors will have to reconstruct more images, while some will have to regenerate less images
- Hence, latency will be induced

System Design - Functionality

Improvement in DIRAS: Load Balancing

- Incorporation of load balancing

System Design - Functionality

Improvement in DIRAS: Load Balancing

- Incorporation of load balancing
- After incorporating load balancing, leader selection algorithm becomes:

$$Leader = j, \quad \text{s.t.} \quad \min_{j} \quad F_{ij}$$

$$F_{ij} = D_{ij} + \beta_j \times \max_{j} \quad D_{ij}$$

$$\text{where} \quad D_{ij} = \sqrt{(x_i(I_i) - x_j)^2 + (y_i(I_i) - y_j)^2}, \quad \text{j} = 1, ..., \textbf{MN}$$

β_j is the total number of images regenerated by that monitor

System Design - Functionality

Improvement in DIRAS: Load Balancing

- Incorporation of load balancing
- After incorporating load balancing, leader selection algorithm becomes:

$$Leader = j, \quad \text{s.t.} \quad \min_{j} \quad F_{ij}$$

$$F_{ij} = D_{ij} + \beta_j \times \max_{j} \quad D_{ij}$$

$$\text{where} \quad D_{ij} = \sqrt{(x_i(l_i) - x_j)^2 + (y_i(l_i) - y_j)^2}, \quad \text{j} = 1, ..., \textbf{MN}$$

  β_j is the total number of images regenerated by that monitor
- The term $\beta_j \times \max_{j} \quad D_{ij}$ ensures that the computation burden is distributed evenly in the monitor node network

Implementation

Implementation - Communication Network

- Done on ns-3 network simulator

Implementation - Communication Network

- Done on ns-3 network simulator
- Involves generating matrices, splitting them and sending them to the monitor nodes

Implementation - Communication Network

- Done on ns-3 network simulator
- Involves generating matrices, splitting them and sending them to the monitor nodes
- Executes leader selection

Implementation - Communication Network

- Done on ns-3 network simulator
- Involves generating matrices, splitting them and sending them to the monitor nodes
- Executes leader selection
- **For evaluation of overheads and delays**

Implementation - Image Reconstruction

- Done on Python (using NumPy and cvxpy)

Implementation - Image Reconstruction

- Done on Python (using NumPy and cvxpy)
- Involves reading an image in form of a matrix and adding noise to it

Implementation - Image Reconstruction

- Done on Python (using NumPy and cvxpy)
- Involves reading an image in form of a matrix and adding noise to it
- Apply RPCA and measure its performance

Implementation - Image Reconstruction

- Done on Python (using NumPy and cvxpy)
- Involves reading an image in form of a matrix and adding noise to it
- Apply RPCA and measure its performance
- Apply Matrix Completion with RPCA and measure its performance

Implementation - Image Reconstruction

- Done on Python (using NumPy and cvxpy)
- Involves reading an image in form of a matrix and adding noise to it
- Apply RPCA and measure its performance
- Apply Matrix Completion with RPCA and measure its performance
- Use Euclidean norm to find the similarity between actual and reconstructed matrices

Implementation - Image Reconstruction

- Done on Python (using NumPy and cvxpy)
- Involves reading an image in form of a matrix and adding noise to it
- Apply RPCA and measure its performance
- Apply Matrix Completion with RPCA and measure its performance
- Use Euclidean norm to find the similarity between actual and reconstructed matrices
- **For studying the performance of image reconstruction**

Implementation - Load Balancing

- Done on Python (using NumPy)

Implementation - Load Balancing

- Done on Python (using NumPy)
- Involves generating random coordinates by the monitors and sensors

Implementation - Load Balancing

- Done on Python (using NumPy)
- Involves generating random coordinates by the monitors and sensors
- Implement load balancing

Implementation - Load Balancing

- Done on Python (using NumPy)
- Involves generating random coordinates by the monitors and sensors
- Implement load balancing
- **For studying the improvement provided by load balancing**

Implementation - Privacy Analysis

- Done on Python (using NumPy and cvxpy)

Implementation - Privacy Analysis

- Done on Python (using NumPy and cvxpy)
- Involves reading image in form of a matrix and removing rows from it

Implementation - Privacy Analysis

- Done on Python (using NumPy and cvxpy)
- Involves reading image in form of a matrix and removing rows from it
- Applying matrix completion to reconstruct the matrix

Implementation - Privacy Analysis

- Done on Python (using NumPy and cvxpy)
- Involves reading image in form of a matrix and removing rows from it
- Applying matrix completion to reconstruct the matrix
- Use Euclidean norm to find the similarity between actual and reconstructed matrices

Implementation - Privacy Analysis

- Done on Python (using NumPy and cvxpy)
- Involves reading image in form of a matrix and removing rows from it
- Applying matrix completion to reconstruct the matrix
- Use Euclidean norm to find the similarity between actual and reconstructed matrices
- **For studying the amount of information revealed by a given number of rows of a matrix**

Evaluation

Evaluation - Communication Network

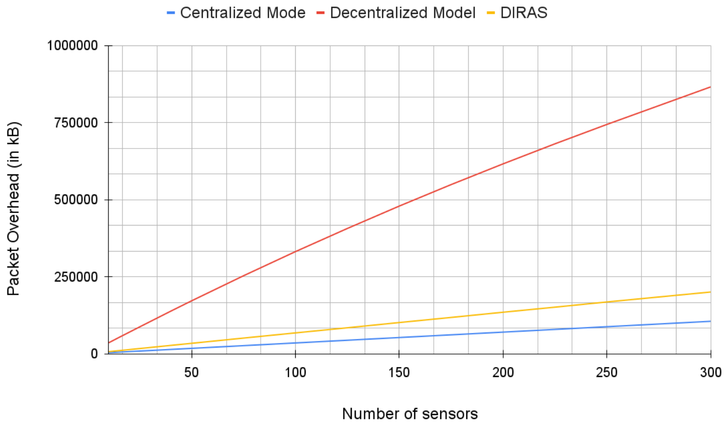

Figure: Variation in packet overhead with change in number of sensor nodes and comparison with centralized and decentralized models

Evaluation - Communication Network

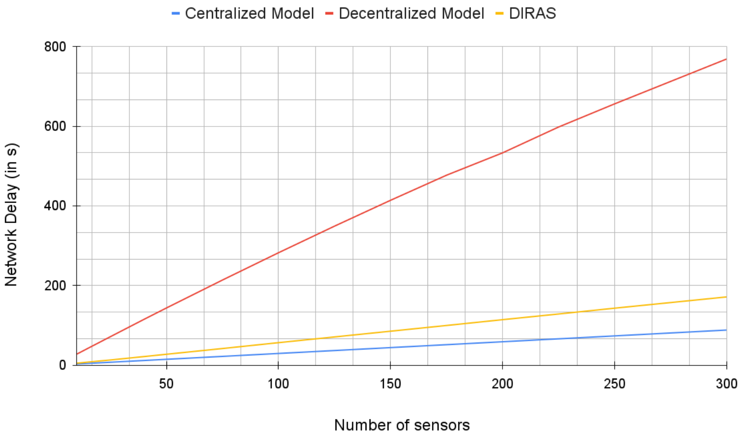

Figure: Variation in delay with change in number of sensor nodes and comparison with centralized and decentralized models

Evaluation - Communication Network

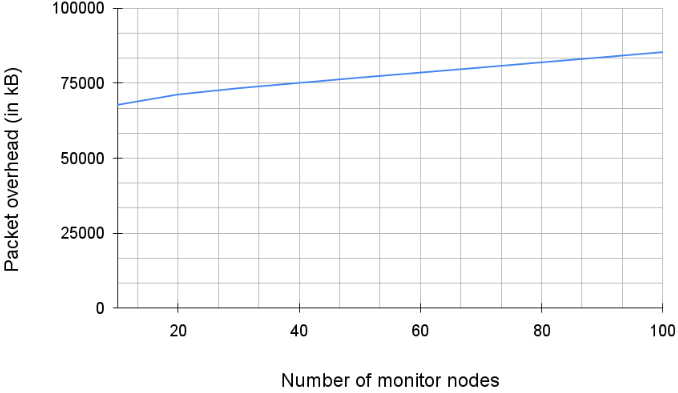

Figure: Variation in packet overhead with change in number of monitor nodes

Evaluation - Communication Network

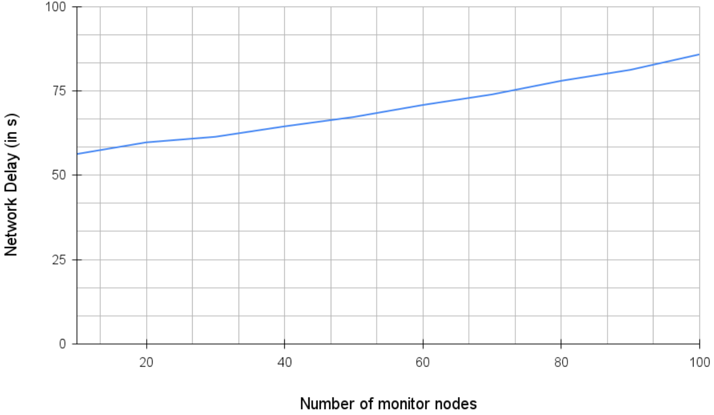

Figure: Variation in delay with change in number of monitor nodes

Evaluation - Image Reconstruction

(a) (b) (c)

Figure: Denoising by RPCA - (a) Actual Image (b) Image with noise (c) Reconstructed image

Evaluation - Image Reconstruction

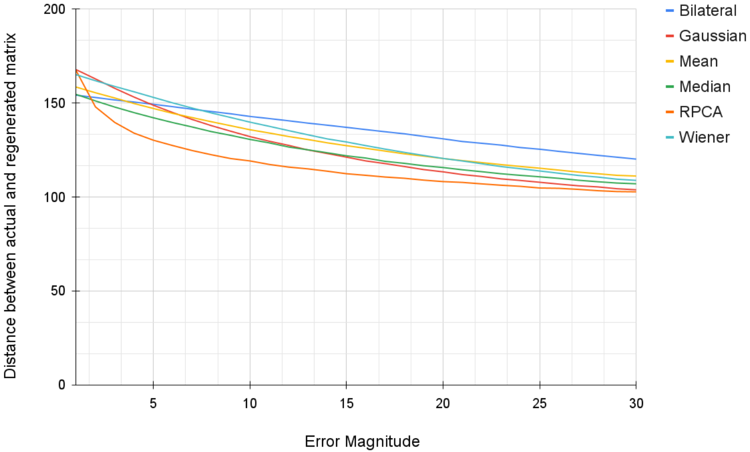

Figure: Performance of DIRAS in case of false data injection and its comparison with other filters

Evaluation - Image Reconstruction

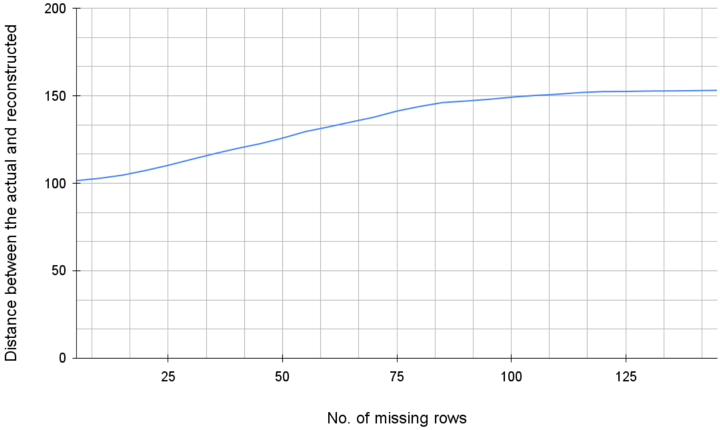

Figure: Performance of DIRAS in case of packet dropping and false data injection

Evaluation - Load Balancing Analysis

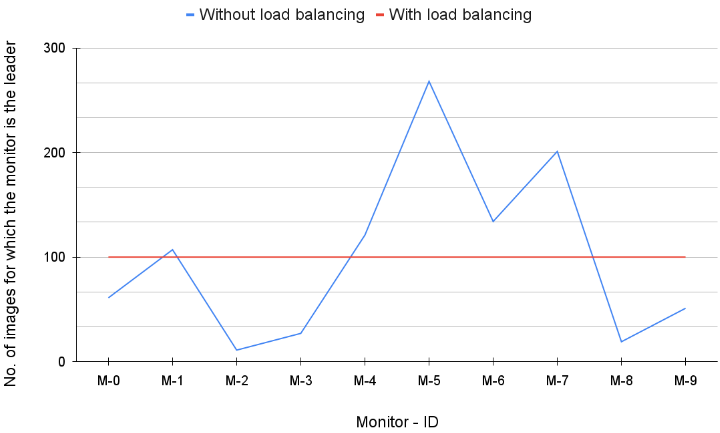

Figure: Improvement in performance because of load balancing based on the number of images for which each monitor node has been chosen as the leader

Evaluation - Privacy Analysis

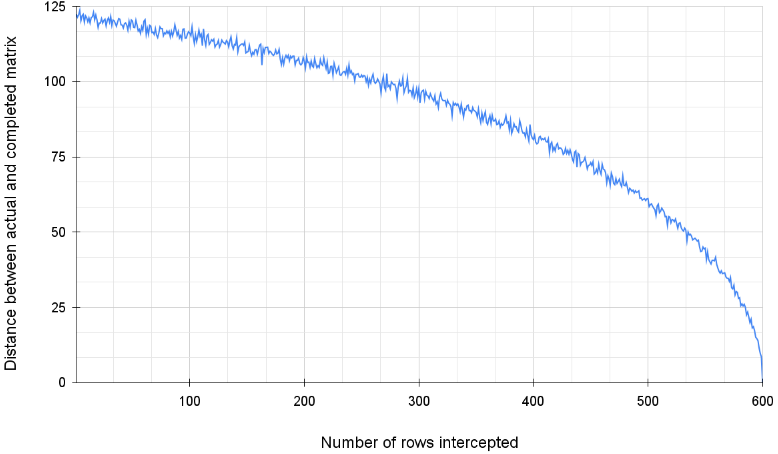

Figure: Privacy provided by DIRAS based on distance between the actual matrix and the matrix reconstructed after applying matrix completion

Discussion

Discussion - Scalability

- Latency and bandwidth consumption very close to centralized model

Discussion - Scalability

- Latency and bandwidth consumption very close to centralized model
- Splitting - reduces the size of data to be sent

Discussion - Scalability

- Latency and bandwidth consumption very close to centralized model
- Splitting - reduces the size of data to be sent
- Bandwidth consumed per channel reduced; thus latency reduced

Discussion - Scalability

- Latency and bandwidth consumption very close to centralized model
- Splitting - reduces the size of data to be sent
- Bandwidth consumed per channel reduced; thus latency reduced
- Distributed architecture - task distributed among the nodes

Discussion - Privacy

- Splitting - reduces the amount of data transferred in a packet

Discussion - Privacy

- Splitting - reduces the amount of data transferred in a packet
- Lesser information implies an adversary gets less information from a packet

Discussion - Privacy

- Splitting - reduces the amount of data transferred in a packet
- Lesser information implies an adversary gets less information from a packet
- Data splitting - enhances data privacy

Discussion - Privacy

- Splitting - reduces the amount of data transferred in a packet
- Lesser information implies an adversary gets less information from a packet
- Data splitting - enhances data privacy
- For successfully obtaining substantial information - attacker needs to control multiple channels, which is difficult - figure 28

Discussion - Privacy

- Splitting - reduces the amount of data transferred in a packet
- Lesser information implies an adversary gets less information from a packet
- Data splitting - enhances data privacy
- For successfully obtaining substantial information - attacker needs to control multiple channels, which is difficult - figure 28
- To reduce number of rows in a chunk - increase the number of monitor nodes - this increases the latency and bandwidth a little - trade-off between network performance and privacy

Discussion - Security

DIRAS mitigates the possibility of multiple attacks:

Discussion - Security

DIRAS mitigates the possibility of multiple attacks:

- **False Data Injection Attack** [Mo et al., 2010]

Discussion - Security

DIRAS mitigates the possibility of multiple attacks:

- **False Data Injection Attack** [Mo et al., 2010]
  - DIRAS uses RPCA [Candès et al., 2011] for removing the noise from the data matrix

Discussion - Security

DIRAS mitigates the possibility of multiple attacks:

- **False Data Injection Attack** [Mo et al., 2010]
  - DIRAS uses RPCA [Candès et al., 2011] for removing the noise from the data matrix
  - However, the performance is not up to mark

Discussion - Security

DIRAS mitigates the possibility of multiple attacks:

- **False Data Injection Attack** [Mo et al., 2010]
  - DIRAS uses RPCA [Candès et al., 2011] for removing the noise from the data matrix
  - However, the performance is not up to mark
  - Maybe because the image used for analysis is not a low-rank matrix

Discussion - Security

DIRAS mitigates the possibility of multiple attacks:

- **False Data Injection Attack** [Mo et al., 2010]
  - DIRAS uses RPCA [Candès et al., 2011] for removing the noise from the data matrix
  - However, the performance is not up to mark
  - Maybe because the image used for analysis is not a low-rank matrix
- **Packet Drop Attack**: DIRAS deploys matrix completion algorithm based on nuclear norm minimization [Keshavan et al., 2010] for obtaining the missing data

Discussion - Security

DIRAS mitigates the possibility of multiple attacks:

- **False Data Injection Attack** [Mo et al., 2010]
  - DIRAS uses RPCA [Candès et al., 2011] for removing the noise from the data matrix
  - However, the performance is not up to mark
  - Maybe because the image used for analysis is not a low-rank matrix
- **Packet Drop Attack**: DIRAS deploys matrix completion algorithm based on nuclear norm minimization [Keshavan et al., 2010] for obtaining the missing data
- **Denial-of-Service (DoS) Attack** [Liang et al., 2016]

Discussion - Security

DIRAS mitigates the possibility of multiple attacks:

- **False Data Injection Attack** [Mo et al., 2010]
  - DIRAS uses RPCA [Candès et al., 2011] for removing the noise from the data matrix
  - However, the performance is not up to mark
  - Maybe because the image used for analysis is not a low-rank matrix
- **Packet Drop Attack**: DIRAS deploys matrix completion algorithm based on nuclear norm minimization [Keshavan et al., 2010] for obtaining the missing data
- **Denial-of-Service (DoS) Attack** [Liang et al., 2016]
  - Monitor nodes in DIRAS change their coordinates with each passing Δ that reduces the probability of the attack

Discussion - Security

DIRAS mitigates the possibility of multiple attacks:

- **False Data Injection Attack** [Mo et al., 2010]
  - DIRAS uses RPCA [Candès et al., 2011] for removing the noise from the data matrix
  - However, the performance is not up to mark
  - Maybe because the image used for analysis is not a low-rank matrix
- **Packet Drop Attack**: DIRAS deploys matrix completion algorithm based on nuclear norm minimization [Keshavan et al., 2010] for obtaining the missing data
- **Denial-of-Service (DoS) Attack** [Liang et al., 2016]
  - Monitor nodes in DIRAS change their coordinates with each passing Δ that reduces the probability of the attack
  - Attacker can still cause damage to the network in a Δ

Discussion - Security

DIRAS mitigates the possibility of multiple attacks:

- **False Data Injection Attack** [Mo et al., 2010]
  - DIRAS uses RPCA [Candès et al., 2011] for removing the noise from the data matrix
  - However, the performance is not up to mark
  - Maybe because the image used for analysis is not a low-rank matrix
- **Packet Drop Attack**: DIRAS deploys matrix completion algorithm based on nuclear norm minimization [Keshavan et al., 2010] for obtaining the missing data
- **Denial-of-Service (DoS) Attack** [Liang et al., 2016]
  - Monitor nodes in DIRAS change their coordinates with each passing Δ that reduces the probability of the attack
  - Attacker can still cause damage to the network in a Δ
  - Load balancing helps in resolving this issue

Discussion - Security

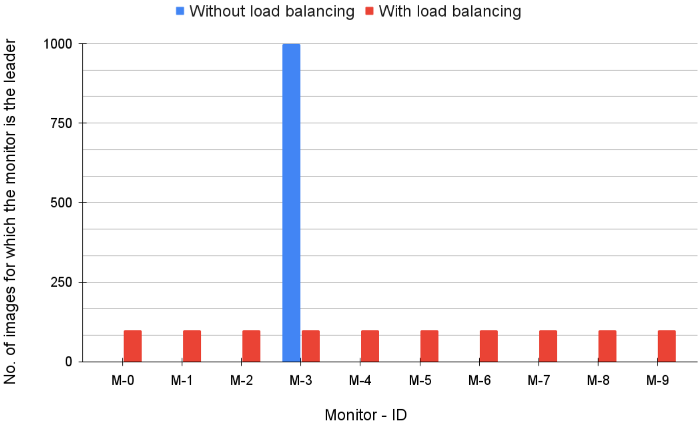

Figure: Mitigation of effect of DoS attacks by load balancing

Future Work

Future Work

Some of these work can be considered for the forthcoming semester:

- Improving the quality of image obtained after reconstruction

Future Work

Some of these work can be considered for the forthcoming semester:

- Improving the quality of image obtained after reconstruction
- Designing the framework when the monitor nodes are untrusted

Future Work

Some of these work can be considered for the forthcoming semester:

- Improving the quality of image obtained after reconstruction
- Designing the framework when the monitor nodes are untrusted
- Designing the framework considering the monitors connected as a graph and not in a p2p network

Future Work

Some of these work can be considered for the forthcoming semester:

- Improving the quality of image obtained after reconstruction
- Designing the framework when the monitor nodes are untrusted
- Designing the framework considering the monitors connected as a graph and not in a p2p network
- Developing a defense mechanism against Sybil attacks [Douceur, 2002]

Future Work

Some of these work can be considered for the forthcoming semester:

- Improving the quality of image obtained after reconstruction
- Designing the framework when the monitor nodes are untrusted
- Designing the framework considering the monitors connected as a graph and not in a p2p network
- Developing a defense mechanism against Sybil attacks [Douceur, 2002]
- Improving the privacy by integrating differential privacy [Dwork, 2008]

Conclusion

Conclusion

- DIRAS is a novel solution for image reconstruction

Conclusion

- DIRAS is a novel solution for image reconstruction
- Distributed - which makes it scalable

Conclusion

- DIRAS is a novel solution for image reconstruction
- Distributed - which makes it scalable
- Provides defense against multiple attacks

Conclusion

- DIRAS is a novel solution for image reconstruction
- Distributed - which makes it scalable
- Provides defense against multiple attacks
- Privacy-preserving

Conclusion

- DIRAS is a novel solution for image reconstruction
- Distributed - which makes it scalable
- Provides defense against multiple attacks
- Privacy-preserving
- Image reconstruction part is still not robust - needs improvements

Conclusion

- DIRAS is a novel solution for image reconstruction
- Distributed - which makes it scalable
- Provides defense against multiple attacks
- Privacy-preserving
- Image reconstruction part is still not robust - needs improvements
- Can be integrated with any system that uses matrix for data storage and processing

References

References I

Atzori, L., Iera, A., and Morabito, G. (2010).
The internet of things: A survey.
Computer networks, 54(15):2787–2805.

Baheti, R. and Gill, H. (2011).
Cyber-physical systems.
The impact of control technology, 12(1):161–166.

Belthangady, C. and Royer, L. A. (2019).
Applications, promises, and pitfalls of deep learning for fluorescence image reconstruction.
Nature methods, 16(12):1215–1225.

Candès, E. J., Li, X., Ma, Y., and Wright, J. (2011).
Robust principal component analysis?
Journal of the ACM (JACM), 58(3):1–37.

References II

Chen, C.-h. (2012).
Signal and image processing for remote sensing.
CRC press.

Douceur, J. R. (2002).
The sybil attack.
In *International workshop on peer-to-peer systems*, pages 251–260. Springer.

Dougherty, G. (2009).
Digital image processing for medical applications.
Cambridge University Press.

Dwork, C. (2008).
Differential privacy: A survey of results.
In *International conference on theory and applications of models of computation*, pages 1–19. Springer.

References III

Guo, Z., Shi, D., Johansson, K. H., and Shi, L. (2016).
Optimal linear cyber-attack on remote state estimation.
IEEE Transactions on Control of Network Systems, 4(1):4–13.

Keshavan, R. H., Montanari, A., and Oh, S. (2010).
Matrix completion from a few entries.
IEEE transactions on information theory, 56(6):2980–2998.

Kurka, P. R. G. and Salazar, A. A. D. (2019).
Applications of image processing in robotics and instrumentation.
Mechanical Systems and Signal Processing, 124:142–169.

Lamport, L. et al. (2001).
Paxos made simple.
ACM Sigact News, 32(4):18–25.

References IV

Liang, L., Zheng, K., Sheng, Q., and Huang, X. (2016).
A denial of service attack method for an iot system.
In *2016 8th international conference on Information Technology in Medicine and Education (ITME)*, pages 360–364. IEEE.

Mo, Y., Garone, E., Casavola, A., and Sinopoli, B. (2010).
False data injection attacks against state estimation in wireless sensor networks.
In *49th IEEE Conference on Decision and Control (CDC)*, pages 5967–5972. IEEE.

Nakamoto, S. (2008).
Bitcoin: A peer-to-peer electronic cash system.
Decentralized Business Review, page 21260.

References V

Nedić, A. and Liu, J. (2018).
Distributed optimization for control.
Annual Review of Control, Robotics, and Autonomous Systems, 1:77–103.

Pasqualetti, F., Dorfler, F., and Bullo, F. (2015).
Control-theoretic methods for cyberphysical security: Geometric principles for optimal cross-layer resilient control systems.
IEEE Control Systems Magazine, 35(1):110–127.

Rong-xiao, G., Ji-wei, T., Bu-hong, W., and Fu-te, S. (2020).
Cyber-physical attack threats analysis for uavs from cps perspective.
In *2020 International Conference on Computer Engineering and Application (ICCEA)*, pages 259–263. IEEE.

References VI

Vibhute, A. and Bodhe, S. K. (2012).
Applications of image processing in agriculture: a survey.
International Journal of Computer Applications, 52(2).

Wang, G., Ye, J. C., and De Man, B. (2020).
Deep learning for tomographic image reconstruction.
Nature Machine Intelligence, 2(12):737–748.

Thank You!