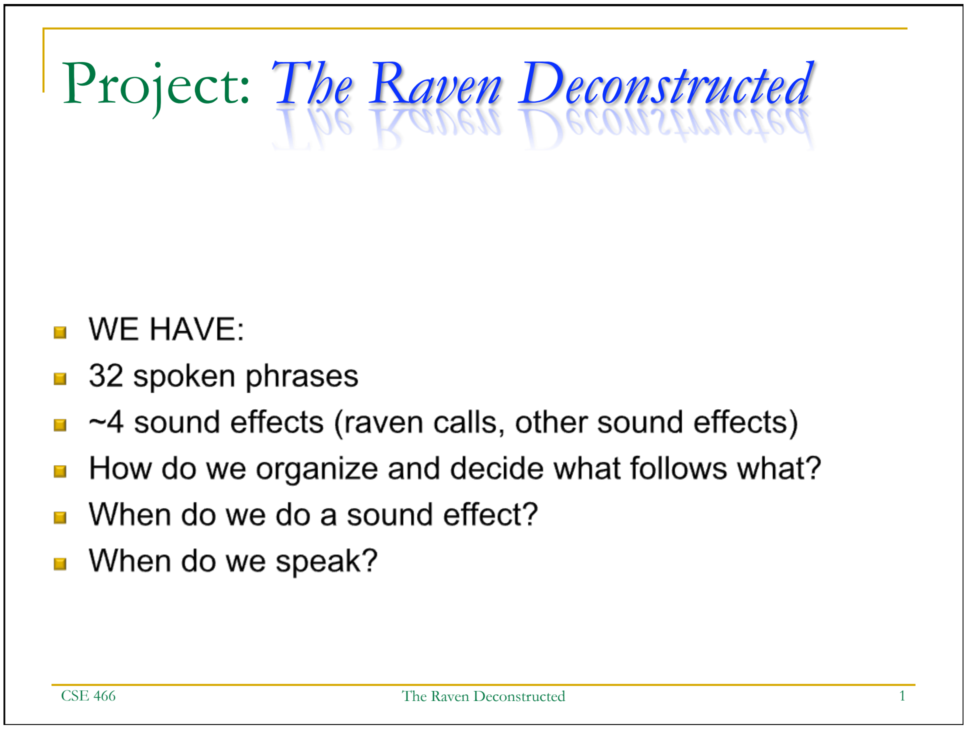
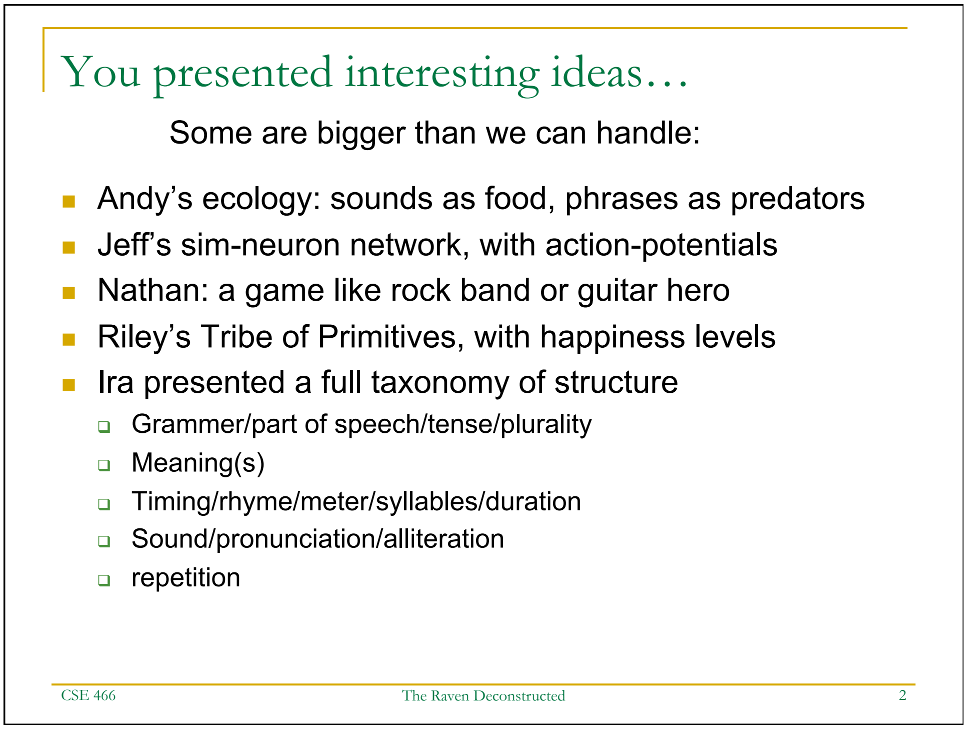


# Project: *The Raven Deconstructed*

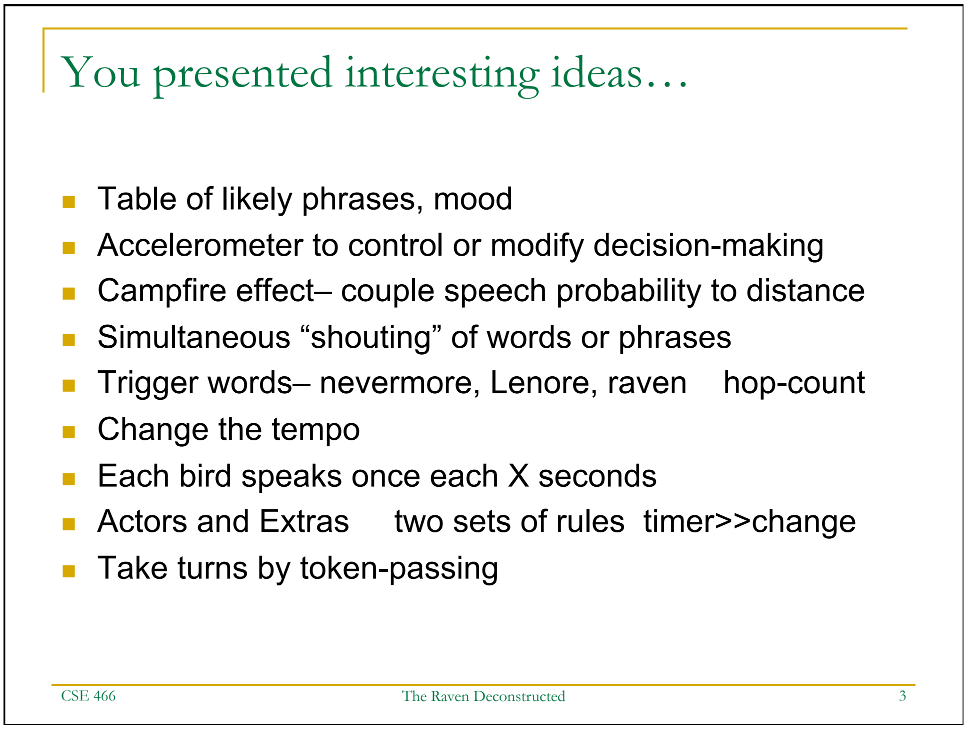
- WE HAVE:
- 32 spoken phrases
- ~4 sound effects (raven calls, other sound effects)
- How do we organize and decide what follows what?
- When do we do a sound effect?
- When do we speak?



# You presented interesting ideas…

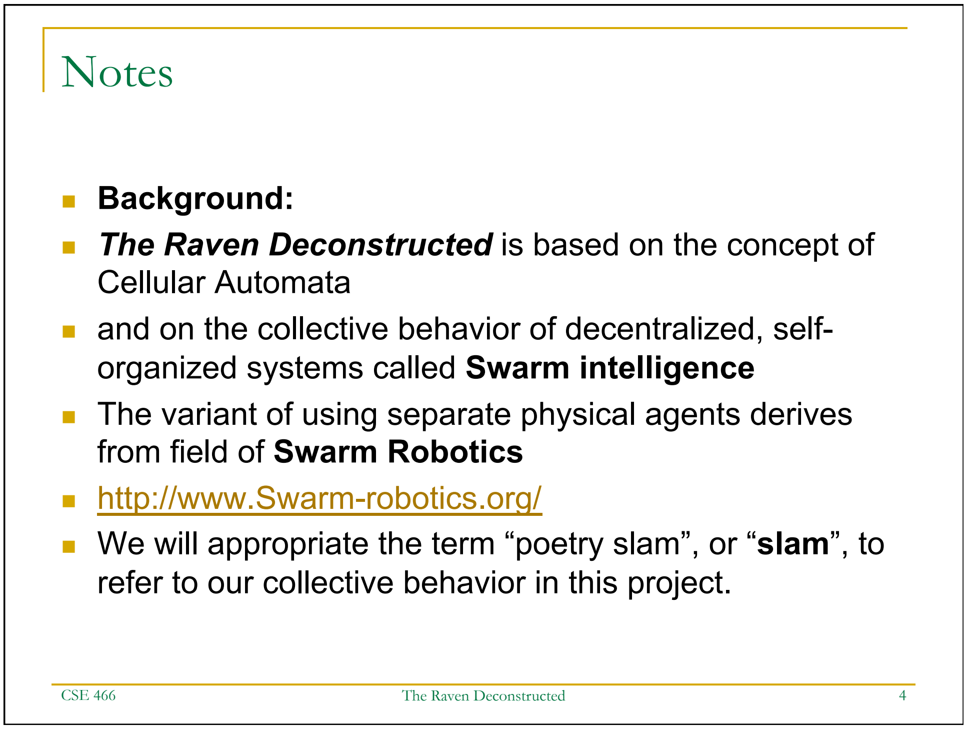
Some are bigger than we can handle:

- Andy's ecology: sounds as food, phrases as predators
- Jeff's sim-neuron network, with action-potentials
- Nathan: a game like rock band or guitar hero
- Riley's Tribe of Primitives, with happiness levels
- Ira presented a full taxonomy of structure
  - Grammer/part of speech/tense/plurality
  - Meaning(s)
  - Timing/rhyme/meter/syllables/duration
  - Sound/pronunciation/alliteration
  - repetition



## You presented interesting ideas…

- Table of likely phrases, mood
- Accelerometer to control or modify decision-making
- Campfire effect– couple speech probability to distance
- Simultaneous “shouting” of words or phrases
- Trigger words– nevermore, Lenore, raven hop-count
- Change the tempo
- Each bird speaks once each X seconds
- Actors and Extras two sets of rules timer>>change
- Take turns by token-passing



## Notes

- **Background:**
- ***The Raven Deconstructed*** is based on the concept of Cellular Automata
- and on the collective behavior of decentralized, self-organized systems called **Swarm intelligence**
- The variant of using separate physical agents derives from field of **Swarm Robotics**
- http://www.Swarm-robotics.org/
- We will appropriate the term “poetry slam”, or “**slam**”, to refer to our collective behavior in this project.

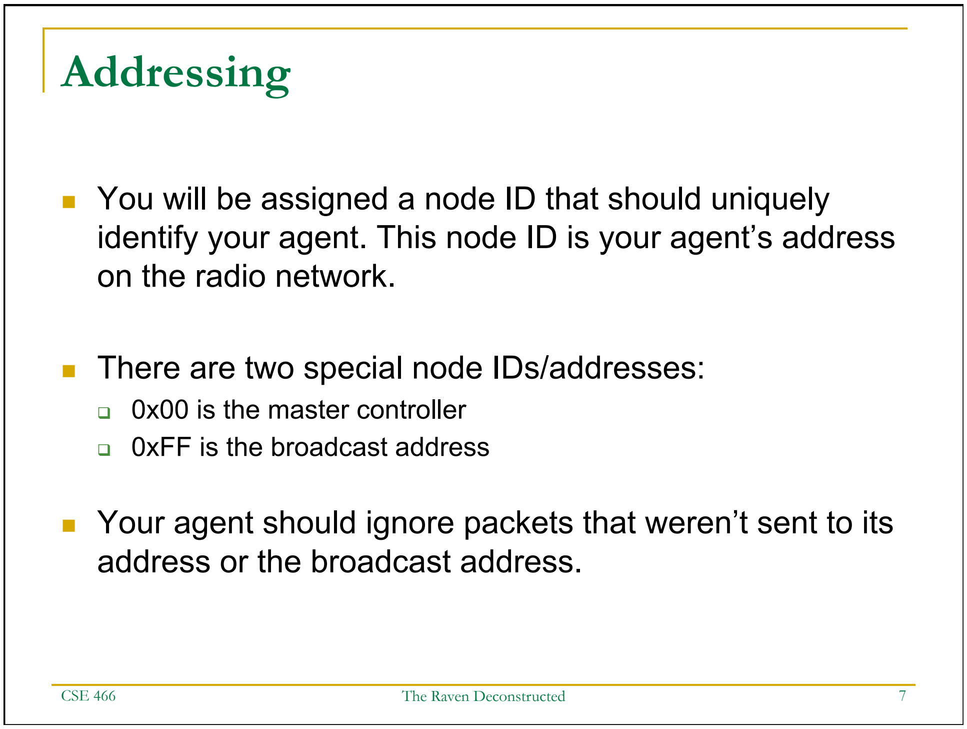
# Description:

- A Slam is a cloud of sound events of related phrases of a poem, in this case “The Raven” by Edgar Allen Poe, as read by James Earl Jones. The choices and of the phrases derive from the execution of a set of common rules, without the intervention of any central control. Therefore, the resultant effect falls under the category of *Emergent Behavior*.
- Slam sounds consist of several types:
- **Phrases:** 32 phrases from the poem, each with length, type (thoughts, subjects, and verbs), and Trigger
- **Raven Sounds:** three distinct raven vocal sounds
- **Storm sounds:** Thunder and rain sounds, two files

# User Interface:

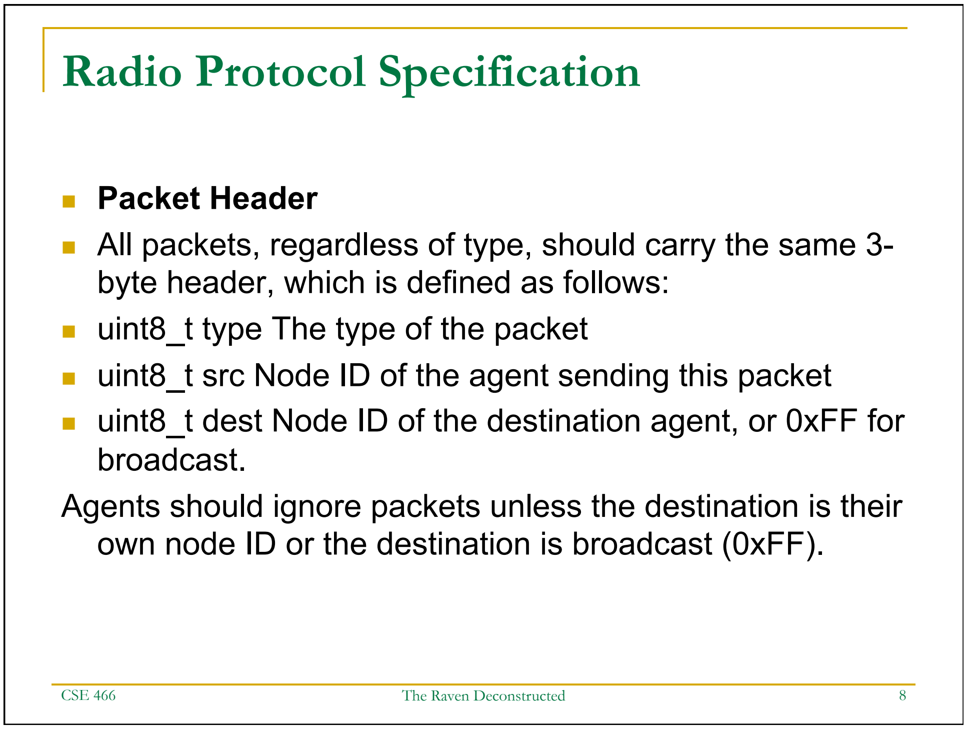
Each pair of partners will implement a GUI interface using the LCD and the rocker switch. This interface will allow for:

- Manual Start/Stop. This will start the process. A global command packet can do this, as well.
- Display: PhraseNum, Transmit Signal Strength, Avg. Received Signal Strength, node ID , other parameters as needed.



## Addressing

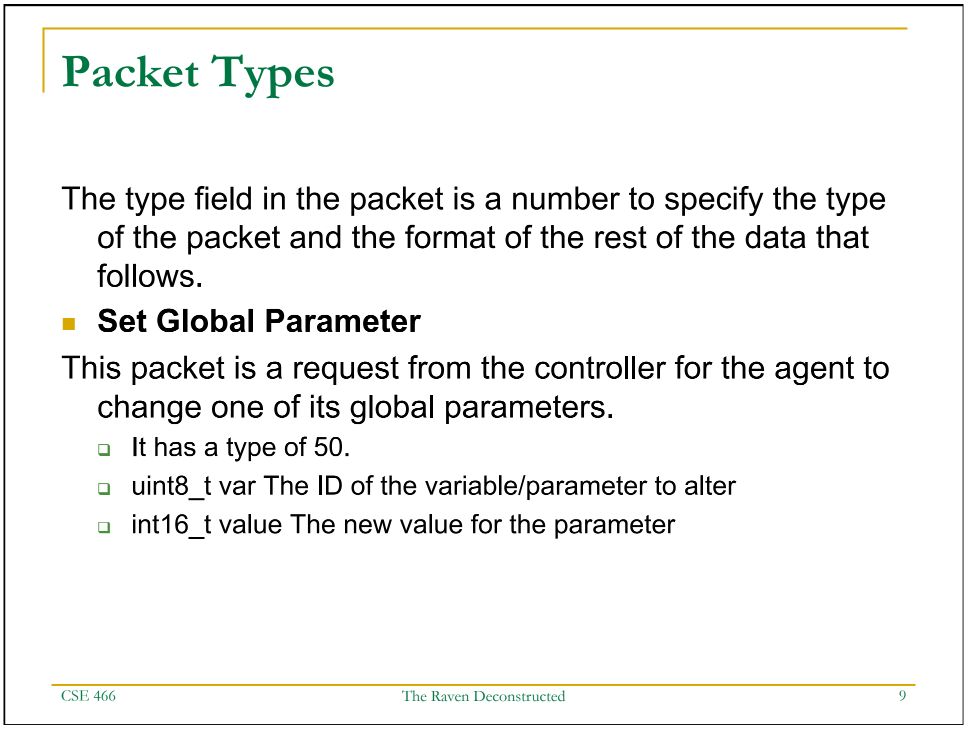
- You will be assigned a node ID that should uniquely identify your agent. This node ID is your agent’s address on the radio network.
- There are two special node IDs/addresses:
  - 0x00 is the master controller
  - 0xFF is the broadcast address
- Your agent should ignore packets that weren’t sent to its address or the broadcast address.



## Radio Protocol Specification

- **Packet Header**
- All packets, regardless of type, should carry the same 3-byte header, which is defined as follows:
- uint8_t type The type of the packet
- uint8_t src Node ID of the agent sending this packet
- uint8_t dest Node ID of the destination agent, or 0xFF for broadcast.

Agents should ignore packets unless the destination is their own node ID or the destination is broadcast (0xFF).



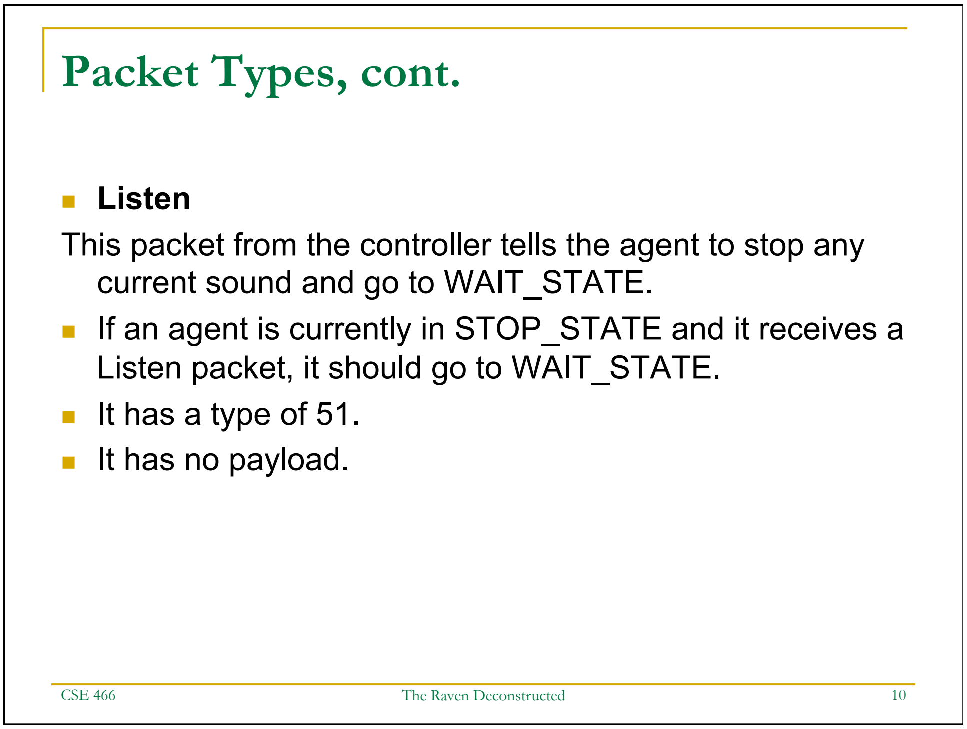
## Packet Types

The type field in the packet is a number to specify the type of the packet and the format of the rest of the data that follows.

- **Set Global Parameter**

This packet is a request from the controller for the agent to change one of its global parameters.

  - It has a type of 50.
  - uint8_t var The ID of the variable/parameter to alter
  - int16_t value The new value for the parameter



## Packet Types, cont.

- **Listen**

This packet from the controller tells the agent to stop any current sound and go to WAIT_STATE.

- If an agent is currently in STOP_STATE and it receives a Listen packet, it should go to WAIT_STATE.
- It has a type of 51.
- It has no payload.

## Packet Types, cont.

- **Do sound**

This packet from the controller tells the agent to stop any sound currently in progress and execute a particular sound.

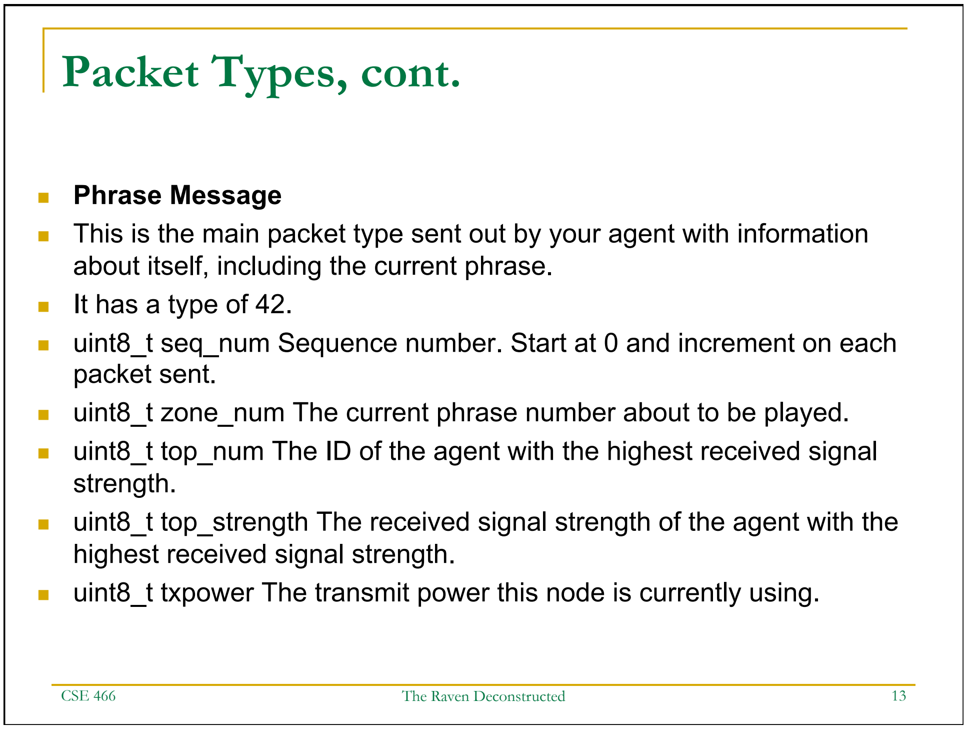
- It has a type of 52.
- uint8_t sound_num The number of the sound to execute.

## Packet Types, cont.

- **Stop**

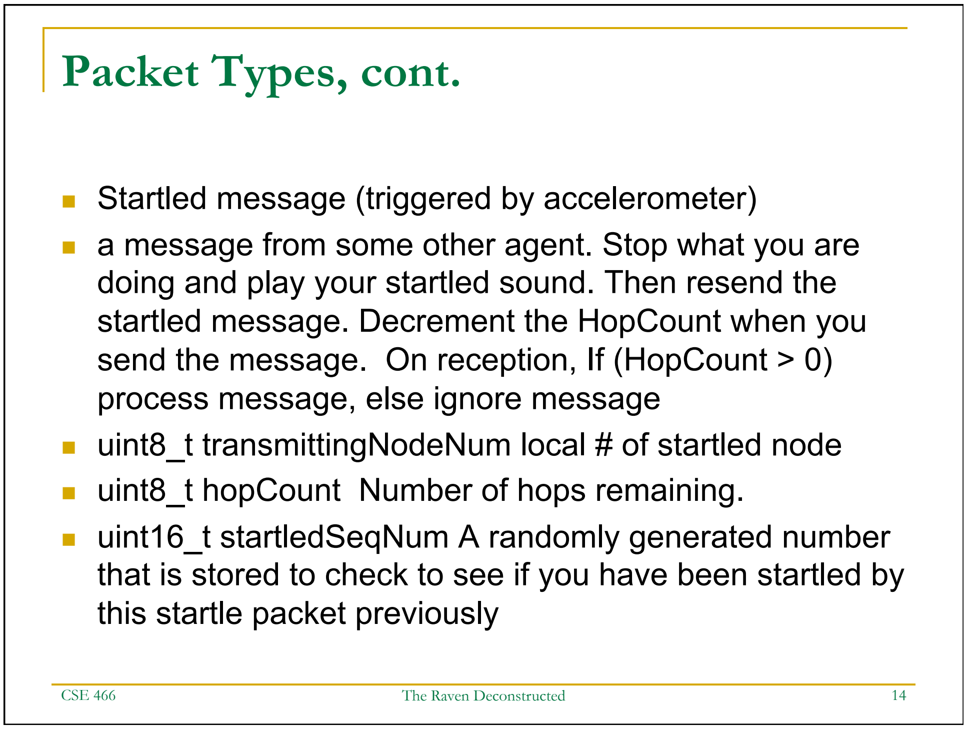
This packet from the controller tells the agent to stop any current sound and go to STOP_STATE.

- It has a type of 53.
- It has no payload.



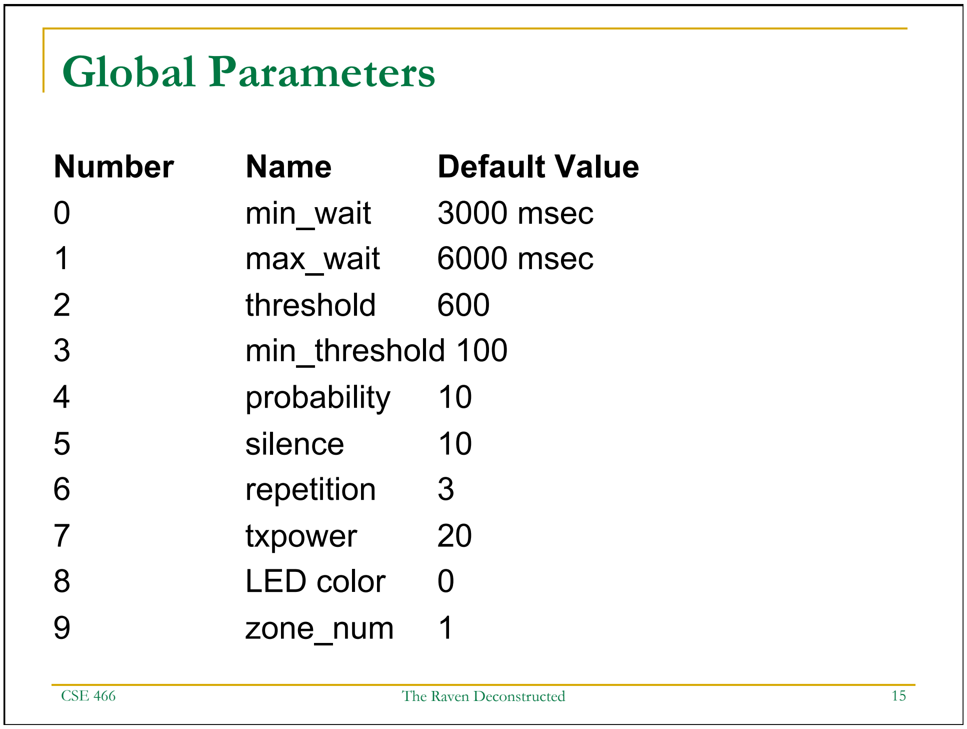
# Packet Types, cont.

- **Phrase Message**
- This is the main packet type sent out by your agent with information about itself, including the current phrase.
- It has a type of 42.
- uint8_t seq_num Sequence number. Start at 0 and increment on each packet sent.
- uint8_t zone_num The current phrase number about to be played.
- uint8_t top_num The ID of the agent with the highest received signal strength.
- uint8_t top_strength The received signal strength of the agent with the highest received signal strength.
- uint8_t txpower The transmit power this node is currently using.



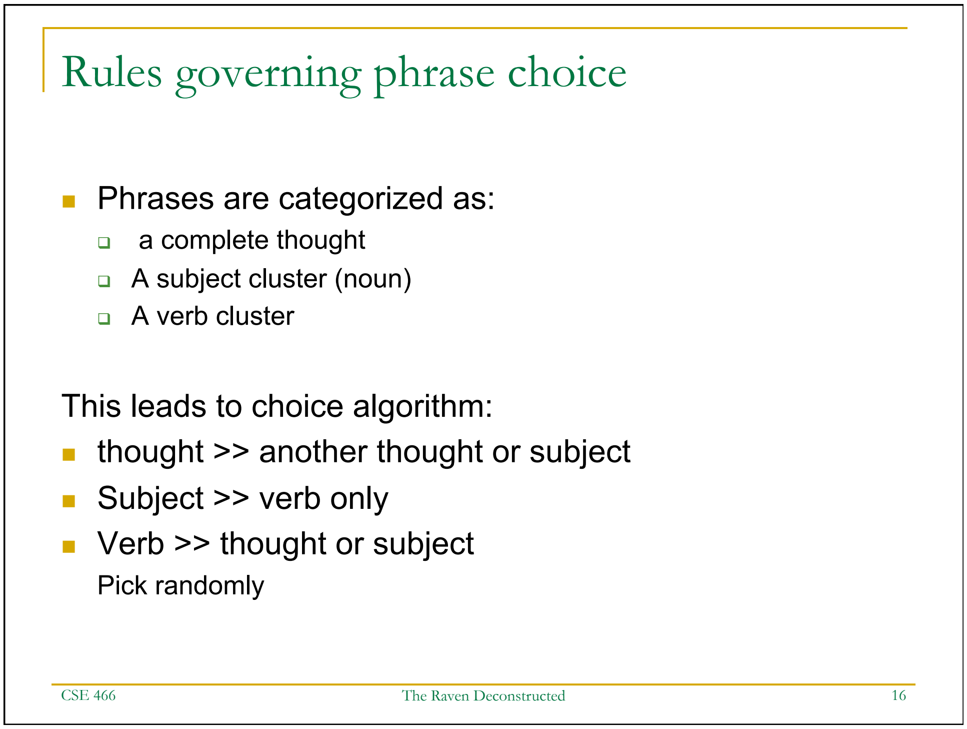
# Packet Types, cont.

- Startled message (triggered by accelerometer)
- a message from some other agent. Stop what you are doing and play your startled sound. Then resend the startled message. Decrement the HopCount when you send the message.  On reception, If (HopCount > 0) process message, else ignore message
- uint8_t transmittingNodeNum local # of startled node
- uint8_t hopCount  Number of hops remaining.
- uint16_t startledSeqNum A randomly generated number that is stored to check to see if you have been startled by this startle packet previously



## Global Parameters

| Number | Name | Default Value |
|---|---|---|
| 0 | min_wait | 3000 msec |
| 1 | max_wait | 6000 msec |
| 2 | threshold | 600 |
| 3 | min_threshold | 100 |
| 4 | probability | 10 |
| 5 | silence | 10 |
| 6 | repetition | 3 |
| 7 | txpower | 20 |
| 8 | LED color | 0 |
| 9 | zone_num | 1 |



## Rules governing phrase choice

- Phrases are categorized as:
  - a complete thought
  - A subject cluster (noun)
  - A verb cluster

This leads to choice algorithm:

- thought >> another thought or subject
- Subject >> verb only
- Verb >> thought or subject

  Pick randomly

# Trigger phrases

- Some phrases trigger a sound response—Amen!
- Procedure: delay length of phrase, then Play trigger sound
- Kinds of trigger sounds:
  - Raven sound 1
  - Raven sound 2
  - Thunder
  - Rain
  - 10-Lenore

# State Machine

- IDLE_STATE
  - Power up, wait for global command or button start
- WAIT_STATE
  - Listen for packet, decide to play a phrase TEST_CCA_PIN()
  - If Carrier clear, transmit **Phrase Message** and go to PLAY_STATE
  - If Trigger Phrase, go to TRIGGER_STATE
  - If Startled, send Startled message, go to PLAY_STATE
- PLAY_STATE
  - Play requested sound file; go to WAIT_STATE
- TRIGGER_STATE
  - Delay phrase_length+-random()
  - Go to PLAY_STATE

What’s missing???

## DISCUSSION…